







Revista de Networking y Programación

www.nexweb.com.ar

NEX # 11 - Septiembre 2004- Precio Argentina 4 $ (recargo interior del País 0.20 $) / Bolivia 10 $ / Chile 1500 $ / México 15 $ / Paraguay 9000 Gs / Uruguay 35 $ / Perú 5.5 ns / Venezuela 2000 bs.

FreeBSD, OpenBSD y netBSD



Cracking Passwords

LC5 y Rainbowcrack

Ethical Hacking: Scanning con nmap

VPNs en LINUX

Super-Free SWAN y Open SWAN

Clusters bajo Windows

©SEVERI

ISSN 1668-5423





Patrocinadores Diamante









21 al 24 de Septiembre, La Rural, Buenos Aires

EXPO COMM / IT ARGENTINA 2004 se transformó ... Por eso hoy se presenta como un evento único en su tipo en todo América Latina, con **+ de 150 Empresas Expositoras** y nuevos **Atractivos Especiales.**



SEMINARIO DE TECNOLOGIA Y MERCADO Las actividades académicas cuentan con la participación de reconocidos especialistas nacionales e internacionales que aportan su visión sobre avances e innovaciones en Comunicaciones e IT y las tendencias del sector.
El esquema de actividades para este año incluye Paneles Temáticos, Presentaciones Magistrales de Keynote Speakers y Desarrollo de Tutorials.



3Com CONVERGENCE CENTER Conozca las soluciones de redes empresariales que pueden cambiar el ritmo de los negocios de su empresa.



CONTACT CENTER EN VIVO Sepa por qué el e-CRM y el Contact Center son factores imprescindibles en las nuevas estrategias de negocios orientados al cliente.

>> 100% tecnología y negocios

Regístrese HOY mismo a la Exposición y Seminario en
http://www.expocomm.com.ar/cortech

Para obtener información adicional sobre el Seminario, comuníquese de Lunes a Viernes de 9 a 19 hs. al +54 (11) 5520 0009 o vía e-mail a expocomm@clienting.com.ar

Organizan







Auspicios Oficiales





PB040823-A

Los menores de 18 años, estando incluso estos, acompañados por un mayor NO PODRAN INGRESAR a la exposición bajo ningún motivo y circunstancia



Editorial

¿Cómo planeamos los contenidos de "NEX IT Specialist"?

Todos los meses debemos decidir, sobre qué temas versarán los artículos a incluir en "NEX IT Specialist". Cada uno del staff editorial, por supuesto, tiene sus preferencias. Si simplemente se adicionasen estas propuestas tendríamos una revista muy distinta a la que está en sus manos.

¿Entonces?

Listemos qué temas podrían ser de interés de nuestros lectores (el IT Specialist): networking (routers, switches), programación (.NET, Java, Mono), Web-design (sobre productos Macromedia, NVU de Open Source); los posibles sistemas operativos: Unix-like (UNIX, BSD (FreeBSD, NetBSD, OpenBSD), Linux), Windows (Win2K, 2003 Server, XP) y otros; las posibles tecnologías: LDAP,ADSL, NAT, protocolos de tuneleo (L2TP/IPsec o PPTP), IPsec, VPNs, SSH, SSL; lo muy nuevo : Wireless (Wi-Fi, WiMAX, Zigbee, Mobile-Fi, ultrawideband); servidores : DNS, DHCP, WINS, Web-servers (Apache, Internet Information Server, IIS de Microsoft), Proxy-servers (ISA server de Microsoft, SQUID); seguridad informática (criptografía, PKI, Ethical hacking, Kerberos, Normas:ISO 17799); temas de educación (IT en universidades, Centros de Capacitación, Certificaciones internacionales); Posibles Modelos de negocio, salida laboral, Cámaras empresariales del mundo IT, grupos de usuarios, eventos, lugares educativos que brindan seminarios de actualización, el movimiento Open Source, e-goverment, e-learning. Y la lista sigue.

Sí, es muy complicado ya que el espectro es muy amplio y los cambios tecnológicos muy rápidos.

Visto de este modo la decisión resulta muy difícil.

Pero no, la receta es muy simple: primero los fundamentos, los pilares sobre lo que se sostiene este mundo IT. Segundo, la historia y sus actores: conocer a los precursores y a sus actores actuales. Finalmente englobar, tratar de obtener "the big picture" de tal o cual temática. Saber dónde se está parado para saber a dónde dirigir los nuevos esfuerzos. Es como cuando juego a "mis ladrillos", LEGO o el mecano (de otros tiempos). Los bloques básicos construyen el castillo.

Decidido qué, la siguiente pregunta es cómo hago esa entrega. Esa entrega deberá ser clara, concisa, de un solo estándar y con referencias para poder ampliar.

De esta forma se concibe "NEX IT Specialist".

Tal como lo dijimos en nuestro anterior número, esperamos contar con Ustedes para tener un feedback a nuestra oferta. No dejen de contactarnos a editorial@nexweb.com.ar

I.T. NEX SPECIALIST

Revista de Networking y Programación

Año 3 - Número 11 - Septiembre 2004

Staff

Preimpresión e Impresión
Impresión: IPESA Magallanes 1315. Capital Federal. Tel 4303-2305/10
Impresión de esta Edición 11.000 ejemplares auditados por IPESA

Distrbución
Distribución en Capital Federal y Gran Buenos Aires: HUESCA-SANABRIA. Baigorri 103. Capital Federal. Tel 4304-3510
Distribuidora en Interior: Distribuidora Austral de Publicaciones S.A. Isabel la Católica 1371.
Capital Federal. Tel. 4301-0701

NEX - Periódico de Networking
Registro de la propiedad Intelectual en trámite leg3038

ISSN 1668-5423

Dirección: Av. Córdoba 657
Piso 12
C1054AAF - Capital Federal
Tel: +54 (11) 4312-7694
http://www.nexweb.com.ar

Indice de Contenidos

NEX IT Specialist # 11




CALENDARIO DE EVENTOS

Septiembre

Expo **"Tecnoar 04"** Se realizará del 2 al 4 de septiembre en el Patio de Madera de la Ciudad de Rosario.
Exposición de Informática y tecnología. Más información: www.tecnoar.org.ar

INFINITION 04- Electronic Entertainment Exposition & World Cyber Games Argentina Preliminary. Se realizará del 3 al 5 de septiembre en La Rural, Capital Federal.
Infinition 04 es el primer mega evento dedicado exclusivamente a la comunidad gamer. Más información: info@gamesandgamers.com.ar

Expo Comm. Se realizará del 21 al 24 de septiembre en La Rural, Capital Federal.
100% Tecnología y Negocios. Más información: www.expocomm.com.ar

Octubre

V Jornadas Universitarias sobre Tecnologías de Internet, "Los Nuevos Paradigmas".

Organiza: La Facultad Regional Buenos Aires de la Universidad Tecnológica Nacional (Departamento Ingeniería en Sistemas de Información, Centro de Estudiantes de Ingeniería Tecnológicos - CEIT- y Secretaría de Cultura y Extensión Universitaria), la Fundación para el Desarrollo del Conocimiento, FUNDESCO.

Auspicia: Cámara Argentina de Comercio Electrónico, CACE, Cámara de Empresas de Software y Servicios Informáticos, CESSI, Universidad Nacional de la Matanza, Universidad de Morón, Polo Tecnológico IT Buenos Aires. JUEVES 14 de OCTUBRE de 2004, LIBERTADOR SHERATON HOTEL, Av. Córdoba y Maipú –

Objetivos: Aunar, en un día de intercambio de experiencias, a la comunidad de intereses e ideales de aquellos universitarios y no universitarios que entrevean que las emergentes tecnologías de Internet coadyuvan al mejoramiento de nuestra sociedad.

1º Congreso Interinstitucional de Tecnología Educativa- 2º Congreso Institucional de Tecnología Educativa- UTN.
Se realizará del 18 al 21 de octubre.
Más información: www.utn.edu.ar

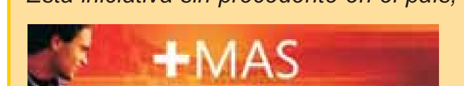
Links de Eventos, Actividades de Entrenamiento y Seminarios del mundo de IT.

http://www.mug.org.ar/Eventos/default.aspx
http://www.linux.org.ar/modules/tinyevent/
http://www.mmug-ar.com.ar/noticias/index.html
http://www.desarrolladoras.com/eventos.html
http://msevents.microsoft.com/cui/default.aspx?culture=es-ar
http://www.intel.com/espanol/events/index.htm?iid=espanolHPAGE+header_trainingevents&
http://www.sun.com/aboutsun/media/
http://www.ibm.com/news/ar/events/
http://www.palermo.edu.ar/eventos/index.html
http://www.cema.edu.ar/conferencias/
http://www.canal-ar.com.ar/
http://www.uade.edu.ar/FRSET_HOME.asp?vPAG=HOME.asp&vMenuDer=Actividades/Home_news.asp&CKtop=100&CKizq=100
http://www.cessi.org.ar/main_sp.htm
http://www.copitec.org.ar/cursos/curso.htm
http://www.novell.com/events/
http://www.cortech.com.ar/gen/sem.htm
http://www.solar.org.ar/rubrique.php3?id_rubrique=9
http://www.infobae.com/interior/detalletecno.php?tipo=2
http://www.informatica.clarin.com/suplementos/informatica/ultimo/index.html
http://www.lanacion.com.ar/Archivo/IndexSeccion.asp?publicacion_id=11964&categoria_id=432
http://www.preteco.com/detalle.php?IDSECCION=98
http://www.cisco.com/ar/ee/index.shtml
http://www.oracle.com/webapps/events/Events.jsp?country=AR

Si quiere recomendarnos algún sitio de interés, escríbanos a eventos@nexweb.com.ar

Microsoft lanza el programa +MAS, para capacitar y dar empleo calificado en tecnología

Esta iniciativa sin precedente en el país,



otorgará en una primera etapa, capacitación tecnológica gratuita para 300 personas y creará más de 100 pasantías o empleos altamente calificados.

Microsoft de Argentina, en conjunto con los polos tecnológicos de Buenos Aires, Tandil, Córdoba, Rosario, Bahía Blanca y la Cámara de Empresas de Software y Servicios Informáticos (CESSI), anunció hoy la realización del "Plan +MAS", a través del cual se entregarán a estudiantes y desarrolladores 300 becas de estudio para participar de un programa de capacitación presencial en .Net. Los asistentes que aprueben los exámenes del curso tendrán, además, la posibilidad de postularse para más de 100 pasantías que empresas clientes y socios de negocio de Microsoft de Argentina otorgarán.

El programa está dirigido a estudiantes universitarios de carreras de Sistemas o relacionadas con tecnologías y programación, así como a desarrolladores interesados en capacitarse y mantenerse actualizados.

Más de 300 jóvenes podrán capacitarse y actualizarse tecnológicamente sin costo. Los asistentes al programa de capacitación podrán postularse, a través de un sitio web (a partir de septiembre en www.empleos-net.com), a búsquedas para cubrir posiciones laborales y ser contactados por empleadores potenciales en base al perfil definido.

Las 100 pasantías o empleos altamente calificados permitirán a los desarrolladores la oportunidad de participar en proyectos, adquirir experiencia e insertarse en el mercado laboral.

El programa de capacitación se extenderá a lo largo de un año. Cada curso presencial consiste en una clase semanal de 3 horas de duración durante 8 semanas, más 30 horas de estudio on-line con tutores. Se llevarán a cabo en la Universidad Abierta Interamericana (UAI), la Universidad Tecnológica Nacional (UTN), la Universidad Argentina de la Empresa (UADE), la Universidad Nacional del Centro de la Provincia de Buenos Aires (UNICEN), la Universidad Nacional del Sur (Bahía Blanca), el Instituto Tecnológico de Córdoba, y en Microsoft de Argentina.

Asimismo, quienes aprueben los exámenes del curso certificarán con 3 estrellas en el programa "Desarrollador cinco estrellas" de Microsoft, el cual ya ha certificado a más de 1.500 desarrolladores.

Este programa se enmarca dentro de un conjunto de acciones que Microsoft de Argentina desarrolla vinculadas con la transferencia de conocimiento, las cuales implican una inversión aproximada de $ 13.500.000.

Para más información sobre el programa +MAS, se puede visitar el sitio: http://www.microsoft.com/argentina/mas .

Kerber

Este artículo pretende dar un entendimiento de Kerberos: para qué sirve, cómo funciona. La filosofía general de Kerberos. Habiendo entendido esto, podemos leer artículos más técnicos. Está basado en el artículo de Brian Tung (The Moron´s Guide to Kerberos, www.isi.edu/~brian/security/kerberos.html). Recomendamos visitar los enlaces citados en el artículo "Excelente Bibliografía sobre Kerberos?" para obtener más documentación.



¿Qué es Kerberos?

Kerberos es un "servicio de autenticación" desarrollado en el MIT (Massachusetts Institute of Technology). Su propósito es el de permitir a usuarios y servicios autenticarse entre ellos. Esto es, les permite demostrar su identidad unos a otros. El MIT desarrolló Kerberos en los años 80 y permitió su libre distribución con la idea de reemplazar lo que era estándar para autenticarse: "autenticación por afirmación" (authentication by assertion). Autenticación por afirmación funciona del siguiente modo: cuando un usuario corre un programa que accede un servicio en la red, el programa (llamado cliente) "afirma" al servicio que corre en representación del usuario. Y eso es todo. No necesitamos recalcarlo, esto provee un muy bajo nivel de seguridad.

La alternativa de requerir la entrada del password del usuario cada vez que se accede a un servicio de red tiene dos problemas. Primero, es demandante para el usuario y segundo, es inseguro cuando se accede a servicios en una máquina remota (el password deberá viajar sin encriptar). Claramente NO aceptable.

Por lo tanto, Kerberos fue diseñado para eliminar la necesidad de demostrar la posesión de información privada o secreta (el password) divulgando esta información Kerberos mismo. Kerberos se basa en el modelo de distribución de llaves (claves, keys) desarrollado por Needham y Schroeder. Una llave (key) se usa para encriptar y desencriptar pequeños mensajes y es ella misma típicamente una secuencia de bytes corta. Llaves (keys) dan la base para autenticación en Kerberos. Todo lo relativo a Kerberos se puede encontrar en el web site del MIT (www.mit.edu/kerberos/www).

¿Qué hace Kerberos?

¿Cómo permite Kerberos autenticarnos? Del mismo modo que nos autenticamos en la vida diaria. Típicamente mostramos nuestro documento de identidad. ¿Y este, qué muestra? Muestra que alguna repartición del gobierno (Policia Federal, Ministerio del interior) certifica la asociación de nuestra identidad a algún aspecto físico (foto, huella digital, altura, color de ojos) que aparece representado en el documento. Al documento se lo considera imposible de copiar. Otro ejemplo de documento podría ser el carnet de conductor que incluiría el nombre, dirección, fecha de nacimiento, y quizás alguna restricción (uso de anteojos). Finalmente la identificación tiene una duración limitada que aparece como la fecha de expiración.

Notemos que la demostración de la identidad requiere además un número de cosas:

>>La tarjeta no debe estar alterada (cambio de fechas, nombre, foto).

>>Que la persona que realiza la autenticación acepta a quien la emitió como de confianza.

Kerberos trabaja básicamente de la misma manera. Es usado cuando un usuario en la red trata de hacer uso de un servicio de red y el servicio requiere asegurarse que el usuario es quien dice ser. Para esto, el usuario presentará un "ticket" (boleto, etiqueta) que ha sido emitido por el "servicio de autenticación" (Authentication Server, AS) de Kerberos.

El servicio examina el ticket para veri-



ficar la identidad del usuario. Si todo está en regla, es aceptado.

Por tanto, este ticket debe contener información que lo ligue inequívocamente al usuario. Como el usuario y el servicio no se enfrentan cara a cara como en el caso de documentos de identidad, una foto no sirve. El ticket debe demostrar que quien lo muestra conoce algo que solo el usuario conoce, como un password. Aun más, debe haber protecciones contra atacantes que roben el ticket y lo usen un tiempo después.

¿Qué cosas asume Kerberos?

Kerberos asume algunas cosas del entorno en el que vive. Por ejemplo, que los usuarios no harán elecciones pobres de los passwords.

Si un usuario elige una password como "password" o "nada", entonces el atacante interceptará algunos mensajes encriptados y montara un "ataque de diccionario" donde probará diferentes passwords a ver cuál los desencripta. Si lo logra, el atacante podrá simular ser el usuario frente a cualquiera. En forma similar Kerberos asume que las máquinas son más o menos seguras y que sólo las conexiones a través de la red son vulnerables de ser comprometidas. En otras palabras, Kerberos asume que no hay manera para un atacante de posicionarse entre el usuario y el cliente de modo de obtener el password de esta forma.

Los Detalles

Veamos más en detalle cómo funciona Kerberos. Tenemos hasta ahora tres actores: Usuario, servicio a ser accedido y el AS (Authentication Server).

Tanto el usuario como el servicio tienen que tener sus llaves registradas con el AS. La llave del usuario se deriva del password por él elegido. La llave del servicio se elige aleatoriamente (no hay nadie que tipee una contraseña).

Para hacer esta explicación más sencilla, imaginemos que los mensajes se escriben en un pedazo de papel (en lugar de ser digitales) y se "encriptan" encerrándolos en una caja fuerte usando una llave (el mecanismo de la cerradura es el algoritmo). En este mundo de "cajas fuerte" los llamados "principales" (usuario y servicios) son creados en el AS y se registra una copia de sus llaves en el AS.

1.El usuario envía un mensaje al AS: "Yo, "E.L. Usuario", quisiera hablar con "Server Tal"."

2.Cuando el AS recibe este mensaje, hace 2 copias de otra nueva llave. Esta se llama "Session key". Se usará para el intercambio directo entre el usuario y el servicio.

3.Pone una de las "Session key" en la caja 1, junto con un pedazo de papel que dice "Server Tal". Cierra la caja con la llave del usuario (recordar que AS tiene una llave de usuario y otra del servicio que fueron registradas). ¿Porqué debe estar ese pedazo de papel? Recordar que la caja es un mensaje encriptado y que la "Session Key" es una secuencia de bytes.Si la caja 1 solo contiene la "Session Key", el usuario no sabría decir si la respuesta vino del AS o si la decriptacion fue exitosa.

4. El AS pone la otra "Session Key" en la caja 2, con un papel con el nombre "E.L Usuario" escrito en él. Cierra la caja con la llave del servicio.

5. Devuelve ambas cajas al usuario

6. El usuario abre caja 1 con su llave, extrae la "Session Key" y el papel que dice "Server Tal".

7. El usuario no puede abrir la caja 2 (ya que fue cerrada con la llave del servicio). En cambio, escribe en un papel la hora actual (timestamp) y la pone en la caja 3.

Cierra la caja con la "Session Key". El envía ambas cajas (2 y 3) al servicio.

8. El servicio abre la caja 2 con su propia llave y extrae la "session key" y extrae el papel con la hora. Esto demuestra la identidad del usuario.

El timestamp se pone en la caja 3 para prevenir que alguien copie la caja 2 (recordar que son mensajes digitales) y la use para simular ser el usuario un tiempo después. Como los relojes no funcionan exactamente en sincronía perfecta se da un pequeño margen (5 minutos es lo típico) entre el tiempo estampado en el papel y el tiempo actual. Además, el servicio mantiene una lista de "authenticators" (caja 3) recientemente enviados de modo de asegurarse que no sean reenviados dentro de la ventana de 5 minutos.

Se preguntarán cómo hace el servicio para abrir la caja 2 cuando no hay nadie "allí" para tipear el password. Lo que sucede es que la llave del servicio no se deriva en un password. Sino que se genera aleatoriamente y es luego guardada en un archivo especial llamado "service key file". Se asume que este archivo es seguro, de modo que nadie puede copiarlo y puede simular ser el usuario legítimo.

En lenguaje Kerberos, la caja 2 se llama "ticket" y la caja 3 el "authenticator". En la realidad el authenticator contiene más información que lo que dijimos más arriba. Parte de esta información es necesaria debido al hecho que este es un mensaje electrónico (por ejemplo, hay un checksum). Puede también haber una llave de encriptación en el "authenticator" para dar privacidad a las comunicaciones entre el usuario y el servicio.

Autenticación del servicio

A veces, el usuario puede requerir que el servicio se autentique. Para esto el servicio toma el timestamp (sello fecha-hora) del "authenticator" (caja 3), la pone en la caja 4, junto con un papel que dice "Server Tal". Lo cierra con la "session key" y lo devuelve al usuario. (Claramente debe agregar algo al timestamp; sino devolvería la caja 3).

> ## *¿De dónde viene el nombre Kerberos?*
>
> El nombre Kerberos viene de la mitología Griega; es un perro con tres cabezas que cuidaba las puertas de Hades. (Hades era un dios pero también la morada de los muertos)
>
> **Kerberos o Cerberus! ¿Cuál es correcto?**
>
> From: Tom Yu <tlyu@MIT.EDU>
>
> "Cerberus" es como se escribe en latín la palabra Griega "Kerberos", y según el OED (Oxford English Dictionary) se pronuncia "serberus", que está en desacuerdo con el griego donde la consonante inicial es una "k". El proyecto Athenas del MIT eligió usar la escritura y pronunciación Griega.
>
> From: Jan Sacharuk <Jan.Sacharuk@cul.ca>
>
> Tom Yu tiene razón, Cerberus es en latín. Sin embargo, el hecho que el OED dice que "c" se pronuncia como "s" es una afección inglesa. En latín, la letra "c" es siempre dura. Así que Cerberus es pronunciado "Ker-ber-ous". La pronunciación de la letra "u" es también un poco diferente, estando entre "us" y "ous".

El Ticket Granting Server (TGS) (Servidor que otorga tickets)

Hay un problema delicado con el intercambio que hemos descrito anteriormente. Se usa cada vez que el usuario quiere contactar un servicio.

Pero, notemos que realizado así se deberá entrar el password (abrir caja 1 con la llave) cada vez. La solución obvia a esto es cachear (guardar en memoria o disco) la llave derivada de la password. Pero ese cacheo es peligroso. Con una copia de esa llave, un atacante podría simular ser el usuario en cualquier momento (hasta que se cambie el password por supuesto).

Kerberos resuelve este problema introduciendo un nuevo agente, llamado "Ticket Granting Server" (TGS). El TGS es lógicamente distinto al AS, aunque ambos pueden residir en la misma máquina. Muchas veces se les refiere colectivamente como KDC (Key Destribution Center, Centro de distribución de llaves) tal como lo llamaron originalmente Needham y Schroeder. La función del TGS es como sigue: Antes de acceder a un servicio regular, el usuario requiere un ticket para contactar al

TGS, tal como si fuese otro servicio regular. Este ticket se llama el "Ticket Granting Ticket (TGT)".

Después de recibir el TGT, cada vez que un usuario desea contactar un servicio, requiere un ticket no al AS, sino al TGS. Aún más, la respuesta está encriptada, no con la llave secreta del usuario, sino con la "session key" que proveyó el AS para usar con el TGS. Dentro de esa respuesta está la nueva "session key" para usar con el servicio regular.

El resto del intercambio ahora continúa como fue descrito anteriormente.

Es parecido a lo que sucede cuando uno visita ciertas empresas. A la entrada uno muestra su documento de modo de obtener una tarjeta de "visitante". Ahora cuando se desea entrar a diferentes secciones de la empresa, en lugar de mostrar el documento una y otra vez (que puede ser robado o se puede caer y perder) uno muestra el ID de visitante, que además es solo válido por corto tiempo. Si fuese robado, uno lo podría invalidar y obtener otro rápida y fácilmente algo que no sucedería con el documento de identidad.

Aclaremos que hay una diferencia en la analogía anterior. El ministerio del interior nos emite el documento de identidad y la empresa el ID de visitante. Estas son entidades separadas física y lógicamente. Por otro lado, en el intercambio del TGT, el AS y TGS son lógicamente distintos pero usualmente físicamente idénticos (en el mismo proceso).

La ventaja que esto nos da es que mientras los password permanecen válidos por varios meses, el TGT es bueno por un periodo corto, típicamente ocho horas. Después de esto el TGT no puede ser usado por nadie: ni el usuario ni un atacante.

El termino "credentials" es usado en literatura de Kerberos en referencia al ticket y "session key" en conjunto. Sin embargo a veces encotrará los terminos "ticket cache" y "credentials cache" usados en forma indistinta.

Diseñando un sistema de autenticación: un diálogo en 4 escenas

Abstract del papel "diseñando un sistema de autenticación: un diálogo en 4 escenas". Es un artículo introductorio a Kerberos, presentado como una obra de teatro en cuatro escenas. Brillante. Lo recomendamos para quien desee comprender Kerberos de un modo divertido.

Este diálogo provee una narración ficticia del diseño de un sistema de autenticación llamado "Charon". A medida que progresa el diálogo, los personajes Athena y Eurípides descubren los problemas de seguridad inherentes a un entorno de redes abierto. Cada problema debe ser abordado en el diseño de Charon, y el diseño evoluciona de acuerdo a esto. Athena y Eurípides no completan su trabajo hasta que el diálogo se cierra.

Cuando terminan el diseño del sistema, Athena cambia el nombre del sistema a"Kerberos". Este nombre coincide por mera casualidad con el sistema de autenticación que fue diseñado e implementado en el MIT bajo el nombre de proyecto"Athena". El sistema "Kerberos" de este diálogo guarda gran similitud con el sistema descrito en *Kerberos: An Authentication Service for Open Network Systems* presentado en el Winter USENIX 1988, en Dallas, Texas.

Se puede encontrar en: <http://web.mit.edu/kerberos/www/dialogue.html>

Autenticación entre Realms (reinos)

Hasta ahora, hemos considerado el caso donde hay un solo AS y un solo TGS, que pueden o no estar en la misma máquina. Mientras que el número de solicitudes sea pequeño no existe problema. Pero cuando las redes crecen, el número de pedidos también y el AS/TGS se vuelve un cuello de botella en el proceso de autenticación. En otras palabras este sistema "NO escalea" (usaremos esta palabra como traducción del concepto en ingles "does not scale"), y esto es muy malo para la idea de un sistema distribuido como Kerberos.

Por tanto es a veces conveniente dividir a la red en "realms" (reinos). Esta división muchas veces se hace basada en divisiones de organización de las empresas, aunque no es necesario que así sea. Cada "realm" tiene su propio AS y TGS.

Para permitir autenticaciones entre "realms", esto es, permitir a usuarios de un "realm" acceder a servicios en otro, es necesario que el "realm" del usuario se registre en un "TGS remoto" (RTGS, remote TGS) en el "realm" del servicio.

Notar que cuando el TGS fue agregado, un intercambio adicional fue agregado al protocolo. Aquí necesitaremos otro intercambio más: primero el usuario contacta al AS para acceder al TGS. Luego contacta al TGS para acceder al RTGS. Finalmente, el usuario contacta al RTGS para poder acceder al servicio deseado.

En realidad esto puede ser peor. En el caso de existir muchos "realms" es muy ineficiente registrar cada "realm" en todos los otros. En cambio se establece una jerarquia de "realms" de modo que para acceder a un servicio en otro "realm", puede ser necesario contactar el RTGS en uno o mas "realms" intermedios. Los nombres de cada "realm" se registran en el ticket.

Kerberos Security en Windows 2000-2003

Cuando se trata de seguridad, Windows (a partir de Win2K) ofrece muchas mejoras sobre Windows NT.

Probablemente, el avance más grande ha estado en el protocolo primario de autenticación del Sistema Operativo. NT LAN Manager (NTLM) ha sido el protocolo primario de autenticación para todas las versiones de NT. Win2K soporta los protocolos de autenticación NTLM y Secure Socket Layer/Transport Layer Security (SSL/TLS). Pero el protocolo primario de autenticación de Win2K es Kerberos 5.
Kerberos consiste en varios sub-protocolos y puede funcionar a través de dominios. El centro de distribución de claves (Key Distribution Center - KDC), el servicio de autenticación de Kerberos, funciona como servicio en cada controlador de dominio (Domain Controler) en Active Directory.

El modelo de seguridad de Kerberos de Win2K tiene varias ventajas sobre los anteriores protocolos de autenticación, incluyendo:

>>Relaciones de confianza transitiva para la autenticación entre dominios.
>>Autenticación eficiente
>>Autenticación mutua de cliente y servidor
>>Autenticación delegada.

Kerberos representa un cambio importante en la manera en que un SO autentica a clientes en una red de Windows. La interoperabilidad del Kerberos deja que Windows autentique a clientes no-Microsoft de otras plataformas (e.g., UNIX), mientras los clientes implementan Kerberos 5. Es sorprendente lo que un perro de tres cabezas puede hacer.

Cómo usar Kerberos

Kerberos es normalmente una infaestructura que subyace ya preparada en los servicios (se dicen Kerberizados). Por ejemplo Windows 2000 lo utiliza en reemplazo de NTLM (LAN manager) para autenticaciones bajo Active Directory (ver nota aparte). Para el usuario por tanto es transparente: él sólo entra su UID y password.

Si alguien desea conocer más detalles los invitamos a leer el artículo de Brian Tung (www.

www.isi.edu/~brian/security/kerberos.htm). Allí podrá entender los pasos que deberá realizar un administrador unix para configurar al AS, TGS, registrar los “principals” en el AS (usuarios y servicios). Y como además deberá kerberizar los servicios (aplicaciones) de modo de utilizarlas con autenticación Kerberos.

Qué encriptación utiliza Kerberos?

En su versión 4 requería el uso de DES (Data Encryption Standard).

En su versión 5 se indica con un identificador (que dice el tipo de cifrado a utilizar).

Entonces se puede usar cualquier técnica de encriptación (cifrado).

Excelente Bilbiografía sobre kerberos

Si Ud es Nuevo en Kerberos recomendamos:

>> Bill Bryant, "Designing an Authentication System: A Dialogue in Four Scenes." <http://web.mit.edu/kerberos/www/dialogue.html>

>> Jeffrey I. Schiller, "Secure Distributed Computing", Scientific American, November 1994, pp 72.

>> J. G. Steiner, B. Clifford Neuman, and J.I. Schiller, "Kerberos: An Authentication Service for Open Network Systems". <ftp://athena-dist.mit.edu/pub/kerberos/doc/usenix.PS>

>> Brian Tung, "The Moron's Guide to Kerberos" <http://www.isi.edu/~brian/security/kerberos.html>

>>La página web de Kerberos de MIT: <http://web.mit.edu/kerberos/www/> tiene muchos links que apuntan a recursos sobre Kerberos. Uno de los mejore tutoriales para Kerberos es el artículo de Jim Rowe, "How To Kerberize Your Site"(Como Kerberizar su sitio), que está disponible en: <http://www.y12.doe.gov/~jar/HowToKerb.html>

Hay también un RFC de Kerberos 5: RFC 1510, disponible en: <http://www.ietf.org/rfc/rfc1510.txt>

IPSEC en Linux: FreeS/WAN y su anunciado final

A mediados de abril del 2004, y tras la llegada de su versión final, la 2.06 junto a novedosas implementaciones como KLIPS para kernel 2.6, el proyecto FreeS/WAN fue declarado finalmente por terminado. El anuncio final dejo sorprendidos a quienes por aquellos tiempos tenían o actualmente tienen corriendo alguna de sus versiones, sobre todo por lo que significa la finalidad del soporte y sus futuras actualizaciones. Como sabemos, la noticia no fue de las mejores que se podrían conocer en el ambiente, pero tendría su parte agradable, y es que si bien un gran proyecto finalizaba, otros, derivados de este y con notables mejoras ya habían nacido…

Decir que el mundo IPSec (IP Security) siempre ha sido complicado para aquellos que se inician o tienen corta experiencia en la tecnología de la información no es ninguna novedad (ver IPsec, un protocolo complejo…), sobre todo para los que necesitan realizar implementaciones sobre plataforma Linux en Kernel 2.0, 2,2 ó mismo sobre 2.4., lo novedoso de esta cuestión es que tras la llegada del Kernel 2.6 junto a un nuevo IPSec Kernel Stack, este mundo no ha dejado de ser menos complejo que el anterior, esencialmente por la confusión que aportan las nuevas cajas de herramientas del entorno de usuario (**Userland Toolkits**) a aquellos que por algún motivo u otro se vieron obligados a migrar o implementar esta nueva tecnología para aprovechar las ventajas que ésta incorpora incluyendo "Opportunistic Encryption", "NAT-Traversal", "certificate handling" y los mayores desarrollos recientes en IPsec para Linux.

Si bien existen alternativas para la creación de Redes Privadas Virtuales en Linux como **vpnd**, resultaba inevitable hasta hace poco tiempo no pensar en FreeS/WAN como la herramienta por excelencia para la creación de dichas redes encriptadas. Tras la llegada de esta gran utilidad, la posibilidad de implementación de una VPN se habría simplificado notablemente, no solo por la gran cantidad de información alojada en su sitio web (www.freeswan.org) sino que principalmente por la simplicidad en sus archivos de configuración, por ello en la actualidad, el mayor porcentaje de implementaciones VPN's realizadas en Linux se encuentran aún montadas sobre FreeS/WAN y sus derivados: **OpenSwan** y **StrongSwan**.

FreeS/WAN, su historia

Allá por el año 1997, John Gilmore (empleado de Sun) junto a uno de los fundadores de Cygnus comenzaron a trabajar sobre un proyecto que desembocaría en corto lapso en el desarrollo de uno de los software más difundidos y usados para autenticar y transmitir datos encriptados de manera segura entre pares: **FreeS/WAN** (Free, Secure Wide Area Network). Este proyecto comprendía la creación de un stack IPsec (Internet Protocol Security) nada más y nada menos que para el sistema operativo que a la postre se convertiría en el más confiable para realizar dicha tarea: Linux. El proyecto se fue obstaculizando por diversos motivos relacionados principalmente a la no aceptación de ningún tipo de código estadounidense ni tampoco el aporte de parches emitidos por sus ciudadanos debido a las regulaciones de exportación criptográficas de los E.E.U.U. en ese entonces. Sin embargo el pro-

Característica	strongswan-2.1.5	openswan-2.1.4
Versión FreeS/WAN	freeswan-2.04	freeswan-2.04
Parche X.509	x509-1.6.3	x509-1.4.8
NAT-Traversal	Linux 2.4 / 2.6	Linux 2.4 / 2.6
AES encryption	Si	No
OCSP	Si	No
CA management	Si	No
XAUTH	No	Si
DPD	No	No

yecto llego a buen puerto brindando un más que seguro y sobre todo estable Stack IPsec para veriones de Kernel 2.0, 2.2, 2.4 y ahora 2.6.

En Septiembre del 2000, Ken Bantoft, actual Business Development de la empresa Xelerance Corporation quien es muy conocido en el ambiente por su gran predisposición y espíritu de ayuda brindada a través de las listas de correo más difundidas sobre el tema, separó FreeS/WAN versión 1.99 en el árbol "**Super FreeS/WAN**" hoy "**OpenSwan**" migrando además los parches más importantes y conocidos (incluyendo parches de connotación política como varios realizados por ciudadanos de los EEUU). Este gran paquete IPsec rápidamente se convirtió en el más popular para Kernels 2.2 y 2.4 derivando además en varios otros proyectos como WOLK (Working Overloaded Linux Kernel), LEAF/bering (Linux Embedded Appliance Firewall) y en varias distribuciones comerciales como Astaro, ImageStream y NIT entre otros.

Tras la llegada del Kernel 2.5, no quedo otra alternativa que generar un nuevo stack IPsec el cual en definitiva debería correr eficientemente bajo la versión 2.6. La posta del desarrollo del nuevo stack fué tomada por David Miller de Red Hat Inc. quien, junto a otros, se encontraba históricamente en conflicto con el proyecto diferente. Pero como es de público conocimiento, para aquel entonces, estos ciudadanos no podrían exportar código criptográfico, por lo que se convertiría en un problema muy serio, no obstante la mayoría era consciente de esto por lo que supusie-

IPSEC, un protocolo complejo...

Como se puede apreciar en el gráfico, IPsec no es un simple protocolo, este requiere interacción entre partes del kernel, como así también *Userland Toolkits* para lograr que trabaje de manera segura. La porción de entorno de usuario maneja la configuración de archivos, llaves/contraseñas y un demonio de comunicación entre pares llamado IKE (Internet Key Exchange). Este demonio negocia con la base de datos de políticas de seguridad (SPD) mediante algún mecanismo predefinido, como por ejemplo llamando a la herramienta setkey directamente o bien vía Kernel socket (*swan).

La SPD contiene todas las políticas de seguridad, las cuales dictaminan que tráfico es encriptado/desencriptado mediante que llaves, y hacia donde es enviado. La Base de Asociación de Seguridad (SAD) contiene todas las SA's activas. En el kernel 2.4, el demonio IKE maneja la tabla de ruteo instalando las rutas necesarias mediante el comando route, o iproute. En kernel 2.6 esto no es requerido ya que el SPD se encuentra directamente en línea en la trayectoria que toman los paquetes a través del stack de red.

FreeS/WAN ya que su opinión se basada en la no aceptación de un código que no pueda ser mantenido por los ciudadanos de los E.E.U.U. por ello, él necesitó algo ron que David no podría escribir el código, ni podría tampoco mantenerlo.

El momento crucial llego de la mano del caso Bernstein (financiado en parte por Juan Gilmore y varios otros vía ambos el FSF, y el EFF) el cual no se caracterizo por ser un triunfo (Ver Caso Bernstein). El resultado es el actual: Sistema "**Notifíquenos**" (notify us system) del BXA (principal ente regulador responsable de los controles de exportación en encriptación, y responsable de la emisión de licencias de exportación) en el cual se debería informar la ubicación en internet, por ejemplo mediante dirección IP, del código fuente en cuestión o bien entregar una copia al momento de la exportación. Muchos desarrolladores de USA consideran al sistema como un triunfo, sin embargo, Gilmore piensa absolutamente lo contrario. Lo cierto es que el tiempo dirá, sobre todo por la gran cantidad de software abierto conteniendo código criptográfico exportado diariamente.

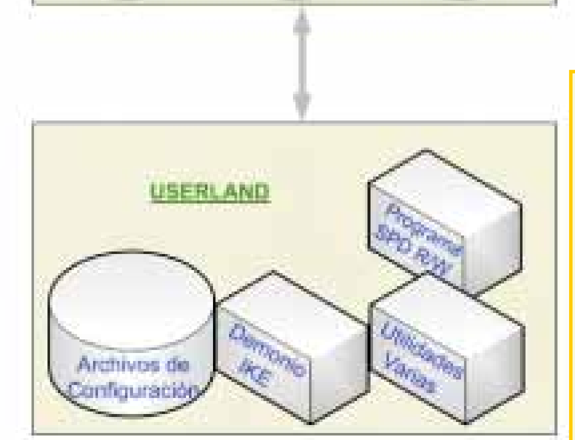


Más tarde, David Miller junto a Derek Atkins y otros eligieron integrar varias piezas del stack USAGI (**U**niver**SA**l play **G**round for **I**Pv6) IPsec (ya que se encontraba funcionando correctamente en IPv6) y *BSD's Kame + Racoon

codebase para Linux liberando a posteriori las userland toolkits. Estas herramientas del entorno usuario ahora conocidas como **ipsec-tools** contienen: por un lado las librerías ipsec mientras que por el otro las herramientas: **setkey** y **racoon** (La primera: para manipulación del *Security Policy Database* (SPD) y del *Security Association Database* (SAD) mientras que la segunda para la negociación automática de llaves en conexiones IPSEC mediante el Demonio *Internet Key Exchange* (IKE))
La incursión de las Toolkits alcanzó al proyecto FreeS/WAN, que se encontraba para ese entonces cercano al final de sus días, momento que no se hizo esperar demasiado, cuando a principios de abril del 2004, su anunciado final se hizo efectivo tras la liberación de una última versión: la 2.06, noticia que pegó fuerte a aquellos administradores de sistemas que se encontraban con la discontinuidad de un producto que les proporcionaba confiabilidad y robustez en las seguras conexiones net-to-net a su cargo. Lo cierto es que las diferencias políticas que separaron a varios de los desarrolladores de FreeS/WAN sentenciaron poco a poco la vida del proyecto y si bien la decisión se prolongó, parecía no tener vuelta atrás, lo que permitió la libertad de toma de decisión con anticipación a migrar a otros desarrollos o bien realizar las nuevas implementaciones con proyectos superiores, entre los que se encuentran dos nuevos nacidos del derivado de su antecesor (FreeS/WAN) y basados en su código: **OpenSwan** (del mismo creador del Super Frees/Wan, Ken Bantoft) y **StrongSwan**, ambos compartiendo la misma herencia y con funcionalidades y características similares. Esta colección de stacks de IPsec que utilizan Pluto como Demonio IKE son actualmente referenciadas como ***swan**.
Las principales características y comparativa de las versiones más recientes de estas soluciones IPsec OpenSource para Linux pueden verse en la tabla.
En conclusión, FreeS/WAN a revolucionado el mundo de las VPN's en Linux demostrando a lo largo de su ciclo de vida que es factible implementar de manera más que eficiente el protocolo de seguridad *IPSec* en la mencionada plataforma, dejando un campo de amplio espectro a los desarrolladores de sus derivados y permitiendo el crecimiento de estos en base a un producto que ganó la confianza tanto de pequeñas como grandes corporaciones, pero sobre todo de aquellos consultores/implementadores que necesitaron generar una vía de comunicación segura entre recursos en Linux.

Martín Sturm
MCSE/MCSA 2000/2003,LPIC-L2,Comptia Linux+ Prof.





Caso Bernstein

Daniel Bernstein poseía importantes ideas sobre criptografía, el arte y ciencia de mantener los mensajes seguros, que él deseaba compartir. En detalle, él deseaba publicar sus ideas e investigación en un foro de discusión en Internet, pero el gobierno de los Estados Unidos le prohibió la exportación de la fuente que describe el sistema de cifrado que él desarrolló, llamado **Snuffle**, sin registrarse como un distribuidor de armas y obtener una licencia emitida por el State Department. El gobierno sostenía que estos ítems se encuentran enumerados en el Listado de municiones de los EEUU y cubieros por International Traffic in Arms Regulations. Pero Bernstein puso en jaque al Gobierno asegurando que "restringiendo el desarrollo de herramientas que permiten anonimato y aislamiento" pone en riesgo las comunicaciones de todos los ciudadanos.
El archivo completo del caso legal: "**Crypto - Bernstein v. US Dept. of Justice**" junto a diversos materiales como archivos con las cartas originales intercambiadas entre Bernstein y varios funcionarios del estado, copia de documentos jurídicos del caso y otros pueden ser consultado en el siguiente acceso: http://www.eff.org/Privacy/ITAR_export/Bernstein_case/

Elementos básicos de criptografía II

Si se quiere ahondar en temas de seguridad informática se deberá tener un claro entendimiento de criptografía.
Los artículos “Elementos de Criptografía I (desarrollado en “NEX IT Specialist #10” de AGOSTO 2004) y II deben ser leídos en el siguiente espíritu: Los puntos en (1-5) dan el basamento, los ladrillos sobre lo que construimos dos herramienta básicas: Firma digital y Key Exchange (6 y 7). En 8 aprendemos otro elemento clave: el certificado y los CA (Certificate Authority). En 9 y 10 entenderemos que se entiende por PKI (dónde armamos el mecano con las piezas anteriores) y daremos algunos ejemplos de donde se usa todo esto. PGP en (10) se incluye para evitar confusiones.

MUY IMPORTANTE:
No confunda ENCRIPTACION de llave pública (también llamada encriptación asimétrica) con INFRESTRUCTURA de llave pública (PKI, Public Key Infrastructure).

6. Firma digital: combinar encriptación asimétrica con hash

Se puede utilizar encriptación asimétrica junto con algoritmos hash para crear una firma digital. Una firma digital actúa como una comprobación de integridad de datos y proporciona una prueba de posesión de la llave privada (autenticación).

Los pasos para la firma digital (autenticación e integridad de datos) son los siguientes:

i) El remitente aplica un algoritmo hash a los datos y genera un valor hash (a veces se lo llama un “message digest”).

ii) Con su llave privada, el remitente encripta (firma) el valor hash. Al valor hash encriptado se lo denomina: “la firma digital del documento”. Es, información basada en el documento y la llave privada de quien firma.

iii) A continuación, el remitente envía al destinatario los datos, la firma digital y el certificado del remitente (en el certificado se envía la llave pública de quien firmó).

iv) El destinatario aplica el algoritmo hash a los datos recibidos y genera un valor hash.

v) El destinatario utiliza la llave pública del firmante para desencriptar el hash encriptado qie le enviaron. Así, compara los hashes para comprobar la firma. Esta comparación de hashes le garantiza que los datos no fueron modificados (integridad) y autentica a quién firmó (autenticación).

Este proceso es transparente para el usuario.

Los algoritmos hash pueden procesar los datos más deprisa que los algoritmos de encriptación asimétrica. La codificación hash de datos también reduce el tamaño de los datos que se van a firmar a una longitud fija y, por tanto, acelera el proceso de firma. Cuando se crea o se comprueba la firma, el algoritmo de llaves públicas tiene que transformar únicamente el valor de hash (128 ó 160 bits de datos).

FUNDAMENTOS DE SEGURIDAD INFORMATICA, PASO A PASO

Paso 1: Elementos de Criptografía I
(ver “NEX IT Specialist # 10 pag. 11)

1. ¿Qué es criptografía?
2. ¿Qué es un un algoritmo?
3. ¿Qué es un hash?
4. Encriptación simétrica: una sola llave
   a). Algoritmos de Encriptación simétrica
5. Encriptación asimétrica: llave-pública y llave-privada
   a). Algoritmos de Encriptación asimétrica

Paso 2: Elementos de Criptografía II

6. Firma digital
7. Envio de llaves secretas (Key Exchange)
8. Certificados: X.509.
9. PKI (Public Key Infrastructure)
10. ¿Dónde se usa todo esto?
11. PGP (Pretty Good Privacy)

7. Intercambio de llaves (Key Exchange): combinar encriptación simétrica con encriptación asimétrica.

Los algoritmos de llaves simétricas son excelentes para encriptar datos de manera rápida y segura. Sin embargo, su punto débil reside en que el remitente y el destinatario deben intercambiar una llave secreta antes de intercambiar datos. La combinación de algoritmos simétricos para encriptar datos con algoritmos de encriptación asimétrica con el fin de intercambiar la llave secreta resulta ser una solución rápida y escalable para el envío de datos encriptados.

Los pasos involucrados en el intercambio de llaves basado en encriptación asimétrica son los siguientes:

i) El remitente obtiene la llave pública del destinatario.

ii) El remitente crea una llave secreta aleatoria.

iii) El remitente utiliza la llave secre-





ta con un algoritmo simétrico para convertir el texto sin formato en texto cifrado.

iv) El remitente utiliza la llave pública del destinatario para encriptar la llave secreta.

v) El remitente envía al destinatario el texto encriptado y la llave secreta encriptada.

vi) Con su llave privada, el destinatario convierte la llave secreta encriptada en texto sin formato.

vii) Con la llave secreta de texto sin formato, el destinatario convierte el texto encriptado en texto sin formato.

8. Certificado y CA (Certificate Authority).

***La llave pública se difunde a cualquiera que la desee tener. La llave privada está en mi posesión y no la difundo. La pregunta es: ¿cómo difundo la llave pública y cómo se garantiza que esa llave pública pertenece a quien* dice ser su dueño (que posee la llave privada asociada) ?**

Un certificado (llamado a veces public-key certificate) es una declaración firmada digitalmente que vincula el valor de una llave pública a la identidad del "*entity*" (persona, dispositivo o servicio) que posee la llave privada correspondiente. Quien firma digitalmente los certificados se llama CA, Certificate Authority. Al firmar el certificado, la entidad emisora de certificados (CA), atestigua que la llave privada asociada a la llave pública del certificado está en posesión del "*entity*" indicado en el certificado.

SINTESIS: un certificado lleva una llave pública y está firmado digitalmente por "alguien" llamado Certificate Authority. TODOS deberemos confiar (trust) en el CA.

Un certificado digital no es un certificado físico con un borde y un nombre con letras llamativas. Es un conjunto de bytes que contienen como mínimo:

i) El nombre de "qué" o "quién" (the entity, "la entidad") está descripto en el certificado. Puede ser una persona (en el caso de e-mails), un servidor (en el caso de https usando SSL).

ii) La llave pública de "la entidad".

iii) Cuándo expira el certificado.

iv) Qué tipo de certificado es. Hay certificados para hacer e-mail seguro, certificados para identificar servidores para IPsec, certificados para hacer seguros a los web servers via SSL.

v) Quién emitió el certificado.

vi) Otro tipo de información que varía con el tipo de certificado.

Esto es lo que hay en un certificado. Pero ¿qué es un certificado?. Es una llave pública e información que identifica a "la entidad" (persona, server o ….) detallados anteriormente y todo esto firmado digitalmente por alguien más. La idea es



que cuando paso mi llave pública, no la paso así nomás sino que la paso firmada digitalmente por alguien (un tercero) en quien "la entidad" y a quien se la paso confían.

Aclaremos partiendo de cero:

1. Genero un par de llaves pública /privada. ¿Pero quién creería que la llave pública que disemino es mía si solo yo tengo la privada asociada?.

Aquí es donde aparece el certificado.

2. Contacto una empresa que otorgue certificados digitales, una Certrificate Authority, CA). Ejemplos: VeriSign, Thawte o Baltimore. Les doy mi llave pública y les pido un certificado. Solo ellos pueden generar un certificado firmado por ellos. Ya que ellos tienen su llave privada. En general estas compañias cobran por el servicio. Puedo yo en mi empresa tener un CA y dar mis certificados que serán creídos por quien confíe en mi CA.

3. La empresa certificadora verifica quién soy yo y emitirá el certificado y lo firmará digitalmente (es decir lo hasheará y firmará digitalmente con su llave privada).

4. Me lo enviarán y yo lo usaré para difundir mi llave pública (por ejemplo, lo instalaré en el software o server apropiado).

Los certificados entonces, proporcionan un mecanismo para establecer una relación entre una llave pública y la entidad que posee la llave privada correspondiente. El formato más común de los certificados utilizados actualmente es X.509. X.509 no es la única forma de certificación. Por ejemplo, el correo electrónico seguro Pretty Good Privacy (PGP) se basa en una forma propia de certificados.

9. Public Key Infrastructure (Infraestructura de llave

PKI en Windows 2000-2003 Server.

Muy probablemente UD sabrá o habrá escuchado que Microsoft ha incluido componentes de la infraestructura PKI en sus sistemas operativos desde Windows 2000. La pregunta es qué puede PKI (public Key Infrastructure) hacer por UD y en qué y cómo lo utiliza con Windows 2003. Recordemos que:

1. PKI es una manera de autenticar basada en estándares e independiente de cualquier sistema operativo.

2. Aunque tiene gran potencialidad, PKI no es usado extensivamente en los sistemas operativos de Windows Server, salvo en algunas pocas aplicaciones.

Para contestar "¿qué puede hacer PKI para uno?", entendamos que:

i) un "certificado" es básicamente un tipo de tarjeta de identificación. Un pasaporte que prueba que uno es quien dice ser con un cierto grado certeza (en realidad que la llave pública asociada al certificado es de la "entidad" descrita en él). Pero no solo sirven para identificar personas. Sirven para identificar servidores y cualquier otra cosa que necesite probar su identidad.

ii) PKI es una infraestructura para administrar el manejo de los certificados. La idea de "cerificados" no es una idea de Microsoft. Las ideas sobre PKI han sido desarrolladas desde hace muchos años en el mundo Unix. Esto es muy bueno ya que es una infraestructura ya muy bien probada y Microsoft se ha basado en las últimas versiones de tecnologías ya muy bien comprendidas. Por ejemplo el formato de los certificados tomado por Microsoft es X.509.

PKI en Windows 2003 Server es aún algo en progreso. Si uno quisiese PKI-izar todo en Windows 2000-2003 sería imposible. Uno, por ejemplo, no puede detener que los Domain Controlers se autentiquen vía Kerberos (ver articulo en "NEX IT Specialist"11, pag. 4 sobre autenticación Kerberos) y comenzar a utilizar PKI. Sí podría setear a sus usuarios para utilizar certificados usando "smart cards". A continuación describimos qué cosas son posibles de utilizar bajo Windows 2003 que usen PKI:

>> Crear y usar certificados para permitir que dos sistemas de diferentes dominios usen IPSec (leer un excelente artículo de IPSec en NEX1). Como sabemos, IPSec es una excelente manera de autenticar y/o encriptar comunicaciones IP entre dos sistemas (dos o más) a través de Internet. Pero esos sistemas necesitan ser capaces de autenticarse y Microsoft ofrece tres opciones: ambos lados comparten una clave secreta (un password), ambos se autentican vía Kerberos (no está mal pero esto funciona sólo si los sistemas pertenecen al mismo forest (bosque). O vía certificados. Usar certificados es la única manera en que dos sistemas que no son miembros del mismo bosque pueden hacerlo vía IPSec.

>> Asegurar el acceso Web. Una de las maneras en que uno puede controlar el acceso a los sitios Web es vía certificados.

>> Crear y usar certificados para asegurar el uso de e-mail dentro de una organización. Uno puede crear certificados para usuarios de e-mail que estos pueden usar bien para autenticarse o encriptar las comunicación vía e-mail. El único problema es que ambos interlocutores deberán aceptar como válidos los certificados creados por "alguien". Por supuesto esto es posible si hablamos de una o dos compañías que son socias. Pero no funcionaría para asegurar el e-mail al mundo. Para ello deberemos "confiar" (to trust) en una fuente única de certificados otorgados por empresas como Thawte.

>> Logons via smart cards: ¿Sirven entonces los certificados para logonearse? Esta es de algún modo la autenticación más común que realiza alguien para acceder a su cuenta. Por supuesto que es posible, pero ¿cómo entramos el "certificado"? En un logon "clásico" uno tipea el userID de la cuenta (nex@nexweb.com.ar o nex) y un password asociado. Entrar además el certificado puede ser muy complejo ya que involucra miles de bytes. Entonces uno podrá logonearse a una cuenta en un dominio con un certificado sólo si éste está almacenado en algo similar a una tarjeta de crédito llamado "smart card". Además, necesitamos que la máquina donde nos logoneamos tenga una lectora de "smart cards". Y eso cuesta.

>> Agentes de recuperacion de EFS (encripted File system): El EFS de Microsoft nos permite elegir personas que pueden desencriptar nuestros archivos en el caso que uno no recuerde el password por ejemplo. Uno designa nuevos Recovery Agents (agentes de recuperación) vía el uso de certificados y PKI.

>> Firmar programas: Como sabrá desde Windows 2000 es posible firmar un dado programa, un programa de instalación de software o un driver. De este modo me aseguro el origen del programa o si ha sido testeado por los laboratorios de Microsoft. El firmado se hace vía certificados.

Como ya dijimos todo esto está en evolución. Y muy probablemente en un futuro cercano podremos llevar una smart card con un certificado personal. Supongamos que tenemos una cuenta en una red Novell, una en un forest de Active Directory de Microsoft y una cuenta de Amazon.com. PKI funcionaría de este modo: entregaríamos una copia de nuestro certificado al administrador de Novell otra al de AD, al de Amazon y cualquier otro donde tengamos una cuenta de usuario. Ellos harán una asociación entre ese certificado y la cuenta en su infraestructura (algo que se llama mapear un certificado a una cuenta). Desde ahí tendríamos un solo password de qué preocuparnos!



pública) (PKI)

¿Cómo monto la infraestructura de manejo y administración de certificados?
Respuesta: PKI.

PKI nos detalla las directivas, los estándares y el software que regulan o manipulan los certificados y las llaves públicas y privadas. En la práctica, PKI hace referencia a un sistema de certificados digitales, entidades emisoras de certificados (CA) y otras entidades de registro que comprueban y autentican la validez de cada parte implicada en una transacción electrónica.

10.¿Dónde se usa todo esto?

Todo lo descrito anteriormente es usado tanto en sistemas Unix como en el mundo Windows.

Por ejemplo a continuación y a modo de ejemplo detallamos lo que puede hacerse con PKI en Windows 2003:

a). Crear y usar certificados para permitir que dos sistemas se comuniquen usando IPsec. Con PKI se pueden autenticar y/o encriptar la comunicación IP entre ellos.

b). Permitir que mi web-browser acceda a un Web-server en forma segura, manteniendo una comunicación encriptada (por ejemplo cuando envío mi número de tarjeta de crédito usando SSL (Secure Socket Layer)).

c). Crear y usar certificados para asegurar los e-mails.

d). Logons usando smart-cards.



Philip R. Zimmermann creador de PGP

Philip R. Zimmermann es el creador de Pretty Good Privacy (PGP) (Privacidad Bastante Buena). Por haber hecho esto, fue blanco de una investigación criminal durante tres años. Esto porque el gobierno americano sostenía que las restricciones para exportación de software de criptografía había sido violado cuando PGP se distribuyó por todo el mundo seguido a su publicación como freeware en 1991. A pesar de su falta de recursos, falta de personal pago, falta de una empresa para respaldarlo y persecución gubernamental, PGP así y todo se transformó en el software para encriptación de e-mails más popular del mundo. Luego que el gobierno retiro los cargos en 1996, Zimmeran fundó PGP Inc. La empresa fue luego adquirida por Network Associates Inc. (NAI) en Diciembre 1997. El permaneció como un Senior Fellow por tres años más. En 2002 PGP fue adquirida por una nueva empresa llamada PGP Corporation, donde Zimmerman ahora realiza tareas de consulta. En la actualidad realiza independientemente tareas de consultaría para una serie de empresas y organizaciones industriales en asuntos de criptografía. Es actualmente un Fellow de Stanford Law School's Center for Internet and Society.



Antes de fundar PGP Inc, Zimmerman era un ingeniero de software con más de 20 años de experiencia, especializándose en criptografía y seguridad de datos, comunicación de datos y sistemas embebidos en tiempo real. Su interés en el lado político de la criptografía, nació de su pasado en asuntos relacionados con políticas militares.

Ha recibido numerosos premios tanto técnicos como humanitarios por su trabajo pionero en criptografía.

Zimmermann recibió su titulo de Licenciado en Ciencias de la computación de la Florida Altantic University en 1978. Es miembro de numerosas organizaciones: International Association of Cryptologic Research, la Association for Computing Machinery y la League for Programming Freedom. Es actualmente el chairman de OpenPGP Alliance y sirve en el Board of Directors for Computer Professionals for Social Responsibility, y en el Advisory Boards for Anonymizer.com, Hush Communications, Veridis y Qualys. (mas info en http://www.philzimmermann.com)

e). Agentes de recuperación de EFS (Encripted File System).

f). Firmar programas.

11. PGP, Pretty Good Privacy

Pretty Good Privacy (PGP) es un paquete de software desarrollado por R. Zimmermann que provee rutinas criptográficas para e-mail y aplicaciones de almacenamiento de archivos.

Lo que hizo Zimmerman es tomar crypto-sistemas ya existentes y protocolos criptográficos y desarrolló un programa que puede correr en múltiples plataformas.

Provee encriptación de mensajes, firmas digitales, compresión de datos y compatibilidad de e-mail.

Los algoritmos que utiliza por default (especificados en el RFC 2440) son : ElGamal y RSA para el transporte de llaves y triple-DES, IDEA y CAST5 para la encriptación de mensajes.

Las firmas digitales se consiguen utilizando DSA para firmar y SHA-1 o MD5 para la computación de los hashes de los mensajes. El programa shareware ZIP es utilizado para comprimir mensajes para transmitirlos y almacenarlos.

La compatibilidad con e-mail se logra con el uso de la conversión Radix-64.



Porqué el PKI de OpenPGP es mejor que el PKI de X.509.

Philip Zimmermann

27 Feb 2001

En la mente de mucha gente, la frase “Public Key Infrastructure” se ha convertido en sinónimo de “Certificate Authority” (CA). Esto es porque en el mundo X.509, el único PKI con que nos encontramos esta construido alrededor del CA. Matt Blaze hizo la siguiente observación: “los CA comerciales nos protegerán de cualquiera al que ese CA se niegue a aceptarle dinero”. Estos CAs están “incluidos dentro” de la mayoría de los browsers, sin que el usuario pueda decidir de confiar en ellos o no.

A lo largo de este artículo, nos referiremos a OpenPGP bajo el standard IETF en lugar de PGP, que es una implementación particular del Standard OpenPGP.

Existe, una Public Key Infrastructure OpenPGP. Pero lo que llamamos una PKI en el mundo OpenPGP es en realidad la amalgama que surge de la suma total de todas las llaves en la población de usuarios, todas las firmas en todas esas llaves, las opiniones individuales de cada usuario de OpenPGP sobre a quién eligen como “introducers” confiables (“trusted introducers”), todos los softwares clientes que corren el modelo de confianzas OpenPGP y realizan cálculos sobre confianzas para cada usuario cliente y los servidores de llaves que en forma fluida diseminan este conocimiento colectivo.

PGP ha crecido por muchos años sin la necesidad de establecer un CA centralizado. Esto es porque OpenPGP usa un sistema de “trusted introducers”, que son equivalentes a un CA. OpenPGP permite a cualquiera firmar la llave pública de cualquier otro. Cuando Alice firma la llave de Bob, ella está “introduciendo” la llave de Bob a cualquiera que confía en Alice. Si alguien confía en Alice para introducir llaves, entones Alice es una “trusted introducer” en la mente de ese observador.

Si yo obtengo una llave que ha sido firmada por varios “introducers” y uno de ellos es Alice, y yo confio en Alice, entonces esa llave está certificada por un “trusted introducer”. Puede estar firmada por otros “trusted introducers”, pero yo no confio en ellos, de modo que no son “trusted introducers” desde mi punto de vista. Es suficiente que Alice haya firmado la llave ya que yo confío en Alice.

Sería aun mejor si dentro de los varios “introducers” de esa llave se incluyeran dos o más personas que yo confiara. Si la llave está firmada por dos “trusted introducers”, entonces estaré más confiado en la certificación de esa llave, ya que es más improbable que una atacante pudiera engañar a dos “introducers” de mi confianza a firmar una llave “trucha”. Las personas pueden ocasionalmente cometer errores y firmar la llave errada. OpenPGP tiene una arquitectura “fault tolerant” (a prueba de fallo) que me permite exigir que una llave esté firmada por dos “trusted introducers” para que sea considerada válida. Esto permite un grado aun mayor de confianza en que la llave pertenece a la persona nombrada en la llave.

Por supuesto, un atacante inteligente podría engañar a dos o más “introducers” no muy sofisticados a firmar una llave pública “trucha”. Pero, eso no es importante en el modelo de confianzas de OpenPGP, ya que yo no confio en “introducers no sofisticados que pueden ser engañados muy fácilmente. Nadie debiera. Uno solo debe confiar en “introducers” honestos y sofisticados que entienden lo que significa firmar una llave, y ejercitarán seriedad en verificar la identidad del poseedor de la llave antes de firmar la llave en cuestión.

Si sólo “introducers” no confiables firman llaves “truchas” nadie será engañado en el modelo de confianzas de PGP. Uno debe decirle al software OPenPGP cliente que “introducers” son de confianza. El software cliente usará ese conocimiento para calcular si una llave está certificada propiamente por un “introducer” confiable mirando las firmas de uno de los “introducers” de confianza. Si la llave no posee firmas de “introducers” que uno le ha dicho al software que confía, el software cliente no considera la llave como certificada y no lo dejará usarla (o por lo menos le indicará a no usarla). Cada uno elige a quien considera un “introducer” de confianza. En muchos casos habrá solapamiento, ya que muchos “introducers” se transforman en confiables para un amplio espectro. Podrían hasta firmar un gran número de llaves como una ocupación full-time. Esos son llamados CAs en el mundo X.509.

No hay nada malo con tener CAs en OpenPGP. Si mucha gente elige confiar al mismo CA para que actúe como un “introducer”, y ellos todos configuran sus propias copias del software cliente de OpenPGP para confiar a ese CA, entonces el modelo de confianzas OPenPGP actúa en idéntica manera que el modelo X.509. De hecho, el modelo de confianza OPenPGP es un “superset” (inversa de subconjunto) del modelo de confianzas centralizado que normalmente vemos en el mundo X.509. No existe ninguna situación en el modelo de confianza X.509 que no pueda ser tratado de idéntico modo en el modelo de confianza de OPenPGP. Pero, OpenPGP puede hacer mucho más, y con una arquitectura “fault tolerant”, y más control del usuario sobre su perpectiva del modelo PKI.

Clusters bajo Microsoft

CLUSTERING PASO A PASO Nro. 3

> Paso 1 NEX 2 Pag. 10 “Beowulf”
> Paso 2 NEX IT Specialist 10 Pag. 16

En este artículo se discutirán las tecnologías que gobiernan y cómo se configura un cluster de servidores bajo el entorno Windows. La utilización de clusters permite aumentar la capacidad de procesamiento o tener tolerancia a fallos (en realidad de alta disponibilidad) dentro de nuestra infraestructura.

Temas expuestos

>> Conceptos generales de clustering
>> ¿Qué ofrece Microsoft?
>> Clusters de alta disponibilidad
>> Clusters de Balanceo de carga
>> ¿Cómo crear un cluster de alta disponibilidad?
>> Un ejemplo
>> Nota sobre ISCSI

Desde los tiempos de Windows NT Server 3.51 estuvo disponible la capacidad de formar un cluster de procesadores. Si bien el realizar un cluster siempre pareció una buena idea para los administradores de sistemas, sólo unos pocos podían en el pasado justificar el costo de realizarlos. Pero, en la actualidad como consecuencia de la caída de precios y los cambios en la tecnología la relación costos/beneficios indica la realización de un cluster como beneficioso. Lo primero que debe entenderse es que existen dos formas de cluster que pueden realizarse: el primero es el cluster de alta disponibilidad (high availability) y el segundo es el cluster de balanceo de carga o NLB (Network Load Balancing), como es llamado por Microsoft.

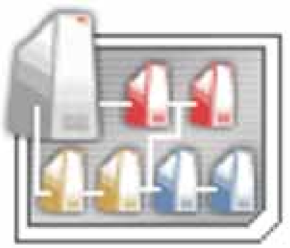
El cluster de alta disponibilidad es aquél en el cual un grupo de servidores trabajan en conjunto y en el caso en que uno de ellos deje de funcionar, otro del conjunto toma el trabajo que ese estaba realizando y continúa con la actividad sin que se produzcan interrupciones. La segunda forma, NLB, es a grandes rasgos definida como múltiples procesadores trabajando juntos como si fueran uno solo para proveer un conjunto de aplicaciones o uno o varios servicios.

El cluster de alta disponibilidad de Windows 2003, disponible en las versiones Datacenter y Enterprise, soporta hasta ocho nodos por cluster. En cambio el cluster NLB, que está disponible en todas las versiones de Windows 2003, soporta hasta 32 nodos por cluster. Pero la limitación que presenta esta herramienta es que Microsoft soporta el uso de sólo uno de estos clusters a la vez. Por ende si se requiere un ambiente con un cluster de alta disponibilidad y además con carga balanceada, deberá recurrirse a soluciones de otras empresas.

Generalmente los ambientes de alto tráfico o las organizaciones grandes que utilizan o desean crear las llamadas “granjas” (farms) de servidores Web y farms de servidores Web de Terminal Services utilizan NLB. Pero si la empresa puede afrontar los costos, se beneficiará del uso del cluster de alta disponibilidad, sobre todo en aquellas empresas en donde una interrupción del sistema puede ocasionar pérdidas importantes de dinero. Ejemplos: la bolsa, sistemas de reservas de compañías aéreas, hospitales, bancos, casinos.). Ésta forma de realizar el cluster también resulta apropiada para servidores de archivos, de impresoras, de correo, y de bases de datos, pero especialmente en servidores de aplicaciones.

El funcionamiento del Servicio de Cluster de Microsoft

Como primer medida el cluster va a necesitar una dirección IP única, y no se puede asignar directamente esa única dirección a todas las máquinas que van a formar el cluster. Por otro lado, Windows cachea la información del disco, incluyendo los contenidos de los archivos, en la memoria RAM para mejorar la performance del disco. Pero el sistema operativo Windows de una máquina no puede “ver” la memoria RAM de las otras del cluster. Por ejemplo, si se escribiese información a un disco compartido desde más de una computadora resultaría corrupto y se perdería la información. Para resolver todos estos inconvenientes y otros relacionados, Microsoft diseñó el Microsoft Cluster Service.

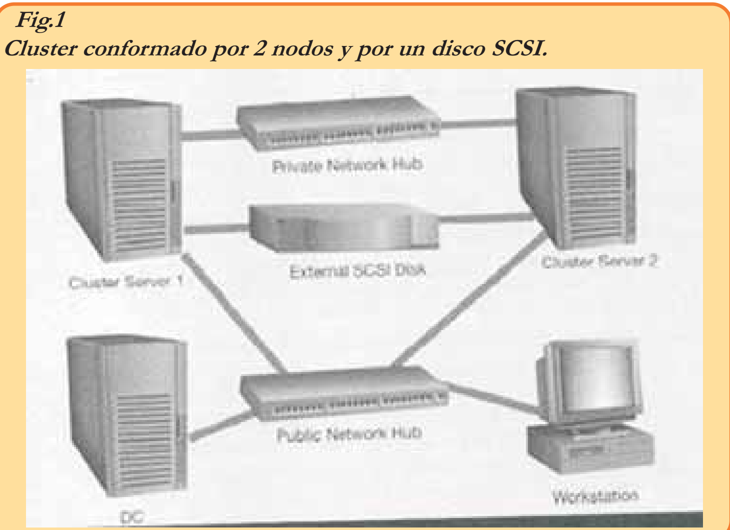


La solución de los problemas mencionados con anterioridad es sencilla con este servicio, ya que el mismo crea una IP fantasma que representa al cluster y que puede ser asignada a cualquiera de sus nodos (recordemos que de todas formas cada máquina debe mantener una IP única propia). Además Microsoft Cluster Service provee un área de disco privada que permite a los nodos intercambiar la configuración del Registry y la información del estado del cluster. Por último también utiliza la red para comunicar el estado de los nodos entre los servidores del grupo.

Remarquemos que por sus características, el Cluster Service de Microsoft es un cluster de alta disponibilidad más que uno con tolerancia a los fallos. La diferencia radica en que según palabras de sus creadores, los ambientes con alta disponibilidad, ante la ocurrencia de un fallo experimentan una pausa momentánea a diferencia de los que poseen tolerancia a fallos que no experimentan ningún efecto al ocurrir la falla.

¿Cómo se crea un cluster? (seguire-

Fig.1
Cluster conformado por 2 nodos y por un disco SCSI.



mos un ejemplo para un cluster de alta disponibilidad: ver figura 1)

En la Fig.1 vemos un grupo de 4 máquinas: una será un DC (Domain Controler), dos ("cluster server1,CS1" y "cluster server2,CS2" que tienen dos placas de red cada uno) formarán nuestro cluster junto a un disco SCSI compartido. Además, una Workstation, WS. CS1, CS2, WS, y DC pertenecen a una subred llamada "Public". CS1 y CS2 a través de su segunda placa están conectados a otra subred llamada "Private". Si instala Windows 2003 server Enterprise Edition en CS1 y 2, tendrá por default el Windows Cluster Service funcionando. WS, representará a un usuario que quiere escuchar un archivo MP3 alojado en el disco SCSI. DC es el Domain Controler de nuestro dominio. Debemos crear en el dominio una cuenta de usuario para el cluster y darle privilegios de administrador local (supongamos CSACC (Cluster Services Account).

Los clusters de alta disponibilidad son creados siguiendo una determinada cantidad de pasos. Primero debe crearse el cluster, y luego asignarle un nombre al mismo. Después se deben agregar los servidores (nodos) al cluster creado. Para finalizar se debe asociar algún recurso con el cluster, como por ejemplo una aplicación o servicio, y de esta forma finalizaría la creación del cluster.

La configuración debe comenzar con uno solo de los servidores que actuarán como nodos de ese cluster. Trabajando con el primer servidor activo deben configurarse todas las propiedades del cluster, para luego ir agregando la cantidad de nodos (servidores) que hagan falta. Para realizar dicha configuración, puede recurrirse a un Wizard, el cual nos irá guiando a través de los distintos pasos de la configuración. Para acceder a ese Wizard debemos ir a Inicio, Herramientas Administrativas y abrir la consola de Administración del Cluster. Luego debe seleccionarse Crear un nuevo Cluster en la ventana de Realizar Conexión de Cluster. A partir de ese momento se irán llenando los distintos campos, como el del dominio al cual se desea conectar ese cluster que se está creando, el propio nombre del cluster y luego el nombre del primer servidor que pertenecerá al cluster. Luego de estos primeros pasos el Cluster Service de Microsoft realiza la comprobación de los datos completados, como puede observarse en la Fig. 2, para luego continuar con la configuración.

El paso que sigue luego de esa comprobación, es seleccionar la IP fantasma de la que hablamos anteriormente, la cual será la IP del cluster, pero que no debe ser ninguna de las direcciones que están siendo utilizadas o serán utilizadas próximamente por los sistemas del dominio. Por último se debe proporcionar el nombre de la cuenta CSACC, la cual se replicará en todos los nodos del mismo con privilegios de administrador local.

De esta forma finalizamos con la creación de un cluster de alta disponibilidad, pero que cuenta con un solo nodo. A continuación, en caso de querer agregar al cluster otro nodo hace falta seleccionar Agregar Nodos al Cluster del menú Realizar Conexión de Cluster, y de esta forma un nuevo Wizard nos guiará hasta finalizar con esa tarea. Nuestro cluster ya tiene dos nodos.

Los Recursos del Cluster ("Cluster resources")

Ahora se debe asociar un recurso (resource) con el cluster. Un resource puede ser un "file share" (archivo compartido), un "print spooler", DHCP, WINS o una aplicación como Exchange o SQL Server. Para crear por ejemplo, un "file share resource" de alta disponibilidad localizado en nuestro disco SCSI, debemos crear usando la herramienta Cluster Administration, un "group", un physical disk resurce, un IP address resource, un "network name" resource y un "file share" resource. Finalmente poner el "file share" resource online. Esto es relativamente directo y se pueden leer los detalles en el link que citamos al final.

Tests y controles posibles

Realizado todo esto podríamos testear nuestro cluster. Desde WS realizaríamos un logon en el dominio para copiar por ejemplo un archivo MP3 al disco SCSI. Comencemos ejecutando el MP3 y testemos que sucede cuando: desconectamos el cable de red de CS1 o CS2 o desenchufemos alguno de ellos.

¿Hubo una pausa en lo que escuchamos?. ¿Hubo diferencia cuando

Fig.2
Vista del Wizard comprobando los datos de creación de un nodo del cluster.

iSCSI

iSCSI es un protocolo que nos permite encapsular los comando SCSI dentro de paquetes TCP/IP. Es una solución de bajo costo para clustering y balance de carga (comparado con una Storage Area Network (SAN). Permite utilizar su infraestructura de red existente para transportar datos SCSI. Microsoft soporta ISCSI en Windows 2003, XP y 2000.

desconectamos a CS1 o CS2? Si desea saber que nodo está activo y cuál pasivo durante los tests use Cluster Administrador o cluster.exe.

Además pueden utilizarse el Microsoft NetMon para analizar el tráfico de la red en cualquiera de los segmentos de la misma, o el System Monitor en los nodos del cluster para realizar una comparación de la performance de cada uno de ellos.

¿Y si deseamos más nodos?

En el caso de querer agregar un tercer o un cuarto nodo, lo mejor es utilizar la tecnología ISCSI (ver nota adjunta). Lo complicado aquí es lo referente a los cables del disco SCSI.

Más detalles:

http://www.microsoft.com/windowsserver2003/technologies/clustering/default.mspx

Rodrigo M. Gonzalez



Una tercera forma de Clusters bajo Windows: HPC

La edición HPC (High Performance Computing) de Windows Server 2003 estará lista para el 2005

Por Carlos Vaughn-O´Connor.

Microsoft confirmó en Junio 2004 que tendrá lista la versión HPC de Windows 2003 Server en la segunda mitad del 2005. Aún no se conocen precios ni demasiados detalles. Solamente se adelantó que se ofrecerá un entorno simplificado para el desarrollo de aplicaciones HPC, instalación de los clusters y su administración.

Computación de Alta Performance (HPC), generalmente se refiere a clusters de docenas a cientos de máquinas, todas procesando una sola instancia de una aplicación. El sector más interesado en este tipo de máquinas es el sector académico y de investigación, pero el modelo computacional está tomando fuerza dentro de ciertos sectores empresariales. Especialmente, en creación de contenidos digitales y aplicaciones de servicios financieros tales como risk-management y equity-management.

El HPC, gira alrededor de clusters para computación en paralelo. Los reportes de IDC muestran una franja del 3 a 4% del mercado de servidores pero está creciendo muy rápido, especialmente dentro del mundo empresarial. La evolución de la idea de clusters se llama “grid computing” o “utility computing” y se está volviendo popular en otros mercados comerciales. Microsoft en ese sentido quiere ganar experiencia para ser uno de los participes de este mercado.

Este anuncio coincidió con la realización de la Internacional Superconducting Conference realizada en Heidelberg, Alemania en Junio de 2004. La lista de las 500 supercomputadoras incluye hoy 287 sistemas basados en tecnología Intel casi 3 veces más sistemas que los 119 del año pasado. Casi todos estos sistemas corren bajo Linux y no Windows. (leer artículo del Dr. Reinaldo Pis Diez en NEX 2, pagina 10 “Clusters Beowulf” y en NEX 11: IT Specialist, pagina 16 : “Open Mosix y Condor”.

Microsoft tiene clientes y partners, entre ellos el prestigioso Cornell Theory Center, trabajando en clusters para HPC sobre versiones de Windows. Pero no puede considerarse a Microsoft como un líder en este segmento. Los clusters corriendo bajo Linux ofrecen performances equivalentes a los mainframes Unix. IDC atribuye el factor precio como muy importante. Con Microsoft se debería pagar una licencia por nodo en uno de estos clusters HPC. Uno podría llegar a tener 1000 nodos. Aún nada anticipó Microsoft sobre el asunto precio. Muy probablemente distribuiría su versión beta entre investigadores y academia para luego concentrarse en la venta a clientes.

La edición HPC de Windows será la primera vez que Microsoft dará soporte al Message Passing Interface (MPI), un estándar que es pilar básico en la industria del HPC. HPC además requiere un stack completo de software entre el hardware y la aplicación que utiliza el cluster. Microsoft aún no ha decidido detalles de este stack. Sí, necesitará entre otras tecnologías un job scheduler y administrador del cluster.

MPI es una interfase multiprocesos para el envío de mensajes, texto e imágenes entre los nodos de un cluster. MPI comprende una librería de subrutinas que manejan la comunicación y sincronización de programas que corren en paralelo.



Ethical Hacking Paso a Paso

Scanning

> **ETHICAL HACKING PASO A PASO**
>
> **Paso 1:** Footprinting. Ver "NEX IT Specialist #10, pag 21.
>
> **Paso 2:** Scanning

Introducción

Como vimos en la edición anterior de *NEX IT Specialist*, aplicando la técnica de *footprinting* se puede obtener una lista de direcciones IP correspondientes a *hosts* y redes usando los utilitarios **`whois`** y **`nslookup`**. Estas herramientas nos permiten obtener información, entre otras cosas, sobre rangos de direcciones IP, servidores DNS y servidores de e-Mail. Con esta información en nuestro poder, ya es posible determinar qué sistemas están "vivos" (encendidos) y alcanzables desde internet utilizando una variedad de herramientas que incluyen *ping sweeps* (barridos de ping), *port scans* (escaneos de puertos), detección de Sistemas Operativos y *automated discovery* (descubrimiento automático).

Ping Sweeps

Uno de los pasos más importantes en el trazado de un mapa esquemático de una red es realizar un barrido de ping sobre rangos de direcciones IP y/o bloques de red para determinar cuáles sistemas están "vivos". Este paso se realiza con herramientas que permiten hacer la misma tarea que el rudimentario **`ping`**, es decir enviar un **ICMP Echo (Tipo 8)** al posible destinatario y esperar obtener un **ICMP Echo_Reply (tipo 0)**, determinando así que el posible destinatario está "vivo".

Aunque existen muchas herramientas disponibles en el mercado, en este artículo nos concentraremos en el uso de **`nmap`**, una poderosísima herramienta desarrollada por ***Fyodor*** (*http://www.insecure.org/nmap/*). En la Figura 1 vemos la sintaxis para realizar un barrido de direcciones IP utilizando el protocolo **ICMP**.

Figura 1 – Resultado de ***ICMP ping sweep*** con **`nmap`**



```
woody:~# nmap -sP 192.168.0.1-254

Starting nmap 3.55 ( http://www.insecure.org/nmap/ )
Host 192.168.0.1 appears to be up.
Host 192.168.0.2 appears to be up.
Host 192.168.0.3 appears to be up.
Host 192.168.0.4 appears to be up.
Host 192.168.0.5 appears to be up.
Host 192.168.0.7 appears to be up.
Host 192.168.0.8 appears to be up.
Host 192.168.0.11 appears to be up.
Host 192.168.0.21 appears to be up.
Host 192.168.0.41 appears to be up.
Host 192.168.0.51 appears to be up.
Host 192.168.0.123 appears to be up.
Host woody (192.168.0.210) appears to be up.
Nmap run completed -- 254 IP addresses (13 hosts up)
woody:~#
```

El modificador **`-s`** permite determinar el tipo de escaneo que se va a realizar (en la Figura 1, la **`P`** adicional indica a **`nmap`** que haga un *ping scan*); las direcciones IP de los posibles destinatarios se pueden especificar individualmente ó utilizando rangos en cualquiera de los octetos (como en la Figura 1), también se pueden especificar bloques de red (usando por ejemplo: **`192.168.0.0/24`** ó **`192.168.0-12.0/25`**). Cuando se hace un barrido de direcciones IP utilizando el protocolo **ICMP**, hay que hacer todo lo posible por evitar las direcciones de *broadcast* (difusión), debido a que estas direcciones tienden a producir ***DoS*** (*Denial of Service*-Negación de Servicio).

El primer problema que se puede presentar, al hacer un barrido de direcciones IP utilizando el protocolo **ICMP**, es que este protocolo esté bloqueado en un *router* ó *firewall* en el borde de una ***DMZ*** (*De-Militarized Zone*-Zona DesMilitarizada). Aquí es necesario hacer un barrido de direcciones utilizando otro protocolo y/o evaluando ciertos puertos conocidos de los posibles destinatarios del barrido. En la Figura 2 vemos la sintáxis para realizar un barrido de direcciones IP sin utilizar el protocolo **ICMP**.

Figura 2 – Resultado de ***TCP probe scan*** con **`nmap`**

```
woody:~# nmap -sP -PT80 192.168.0.1-254

Starting nmap 3.55 ( http://www.insecure.org/nmap/ )
Host 192.168.0.1 appears to be up.
Host 192.168.0.2 appears to be up.
Host 192.168.0.3 appears to be up.
Host 192.168.0.4 appears to be up.
Host 192.168.0.5 appears to be up.
Host 192.168.0.7 appears to be up.
Host 192.168.0.8 appears to be up.
Host 192.168.0.12 appears to be up.
Host 192.168.0.21 appears to be up.
Host 192.168.0.31 appears to be up.
Host 192.168.0.41 appears to be up.
Host woody (192.168.0.210) appears to be up.
Nmap run completed -- 254 IP addresses (12 hosts up)
woody:~#
```

El modificador **`-s`** permite determinar el tipo de escaneo que se va a realizar (en la Figura 2, la **`P`** adicional indica a **`nmap`** que haga un *ping scan*); pero el modificador **`PT80`** le dice a **`nmap`** que haga un TCP probe scan. Así, utilizando el protocolo **TCP** sobre el puerto **80**, se logra que el barrido supere un posible *router* y/o *firewall* debido a que muy probablemente el tráfico sobre el puerto **80** del protocolo **TCP** esté permitido.

Como se puede ver, esta técnica es muy efectiva para superar el escollo del bloqueo del tráfico **ICMP**. También vale la pena reintentar el mismo rango de direcciones utilizando diferentes puertos de protocolos conocidos, por ejemplo: **FTP** (21) **SMTP** (25), **POP3** (110), **RPCBIND** (111), **IMAP** (143), **MSRPC** (135).

Ping Sweeps - Contramedidas

El proceso de detección de un *Ping Sweep* es crucial para determinar si va a ocurrir un ataque, cuándo va a ocurrir y quién lo va a realizar. El principal método de detección de un *Ping Sweep* es utilizar un ***NIDS*** (*Network-based Intrusion Detection System*-Sistema de Detección de Intrusos basado en Red).

Mientras que la detección de los *Ping Sweeps* es crítica, la prevención también hará una contribución substancial. Es recomendable que evalúe el tipo de tráfico **ICMP** que permite circular en su red. Recuerde que existen 18 (dieciocho) tipos distintos de tráfico **ICMP**, **Echo** y **Echo_Reply** son sólo 2 (dos) de ellos.

Port Scanning

Hasta aquí hemos visto cómo se puede determinar qué sistemas están "vivos", utilizando las técnicas de *Ping Sweep* ó de *TCP Probe Scanning*. Habiendo recolectado esta información, ya es posible hacer un *Port Scanning* (Escaneo de Puertos) sobre cada equipo/sistema individual. *Port Scanning* consiste en establecer conexiones **TCP** y **UDP** a un equipo de destino (una posible "víctima") para establecer qué servicios están en ejecución o en estado

Listening (escuchando). Los servicios activos que estén escuchando pueden permitir el acceso no autorizado a usuarios no deseados. Estos usuarios podrían lograr acceso a servidores que están mal configurados o que tienen instaladas versiones de aplicaciones que tienen vulnerabilidades conocidas.

Figura 3 – Resultado de ***TCP SYN scan*** con **`nmap`**

```
woody:~# nmap -sS 192.168.0.4

Starting nmap 3.55 ( http://www.insecure.org/nmap/ )
Interesting ports on 192.168.0.4:
(The 1648 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE
21/tcp    open  ftp
25/tcp    open  smtp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
1025/tcp  open  NFS-or-IIS
1400/tcp  open  cadkey-tablet
1433/tcp  open  ms-sql-s
3372/tcp  open  msdtc
3389/tcp  open  ms-term-serv
MAC Address: 00:08:54:05:F5:32 (Netronix)

Nmap run completed -- 1 IP address (1 host up)
woody:~#
```

Port Scanning – Tipos de Scan

Existen varios tipos de escaneo de puertos, dando una perspectiva distinta de cómo detectar servicios y/o aplicaciones. Asimismo, los diferentes tecnicismos de cada uno de ellos permitirá hacer los escaneos con un mayor o menor grado de sigilo.

>>TCP connect scan: este tipo de *scan* se conecta al puerto de destino haciendo un *Three-Way Handshake* completo. (Pasos 1, 2 y 3 de la Figura A de la pastilla)

>>TCP SYN scan: esta técnica es conocida también como "*half-open scanning*", debido a que solamente se envía un paquete con el *flag* ***SYN*** a la "víctima", si ésta responde con un *flag* ***SYN/ACK*** el puerto esta escuchando, si responde con un *flag* ***RST/ACK*** el puerto esta cerrado. Para evitar el *Three-Way Handshaking*, se envía un paquete con el *flag* ***RST/ACK***, y así se logra que la "víctima" no registre una conexión.

>>TCP FYN scan: con esta técnica se envía un paquete con el *flag* ***FIN*** a la "víctima", y ésta debe responder un paquete con el *flag* ***RST*** para los puertos que estén cerrados.

>>TCP Xmas Tree scan: con esta técnica se envía un paquete con los *flags* ***FIN, URG*** y ***PUSH*** a la "víctima", y ésta debe responder un paquete con el *flag* ***RST*** para los puertos que estén cerrados.

>>TCP Null scan: con esta técnica se envía un paquete que tiene todos los *flags* apagados a la "víctima", y ésta debe responder un paquete con el *flag* ***RST*** para los puertos que estén cerrados.

>>UDP scan: con esta técnica se envía un paquete a un puerto específico de la "víctima", y ésta debe responder un paquete **ICMP Port_Unreachable** para los puertos que estén cerrados. Los únicos problemas de esta técnica son: su falta de fiabilidad y su baja performance.

En la Figura 3 podemos ver a **`nmap`** haciendo un *TCP SYN scan* sobre uno de los destinos "vivos" de la red.

El modificador **`-s`** permite determinar el tipo de escaneo que se va a realizar (en la Figura 3, la **`S`** adicional indica a **`nmap`** que haga un *TCP SYN scan*); es posible especificar el FQDN de la "victima", pero es preferible usar su dirección IP.

Las direcciones IP de los posibles destinatarios se pueden especificar individualmente ó utilizando rangos en cualquiera de los octetos (como en la Figuras 1 y 2, pero hay que hacer todo lo posible por evitar las direcciones de *broadcast* (difusión).

Para los demás tipos de escaneo es necesario usar una **`T`** (*TCP connect scan*), **`F`** (*TCP FIN scan*), **`X`** (*TCP Xmas Tree scan*), **`N`** (*TCP Null scan*) ó **`U`** (*UDP scan*).

Si agregamos el modificador **`V`** (resultando en **`-sSV`**) **`nmap`** tratará de informarnos de la versión de la aplicación y/o servicio que está escuchando en ese puerto.

La cantidad de modificadores que tiene **`nmap`** y sus posibles combinaciones hacen imposible que lo mostremos en este artículo.

Existen además otras herramientas disponibles, un ejemplo es la herramienta **`portqry`** de Microsoft. Es gratuita y se puede buscar y bajar del Knowledge

Figura 4 – Resultado de detección de SO con **`nmap`**

```
woody:~# nmap -O 192.168.0.21

Starting nmap 3.55 ( http://www.insecure.org/nmap/ )
Interesting ports on 192.168.0.21:
(The 1643 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE
9/tcp     open  discard
13/tcp    open  daytime
  :
901/tcp   open  samba-swat
933/tcp   open  unknown
9999/tcp  open  abyss
MAC Address: 00:08:54:06:16:DB (Netronix)
Device type: general purpose
Running: Linux 2.4.X|2.5.X|2.6.X
OS details: Linux 2.4.0 - 2.5.20, Linux 2.4.18 - 2.6.4 (x86)

Nmap run completed -- 1 IP address (1 host up)
woody:~#
```

Base del sitio de Microsoft (www.microsoft.com). Otro ejemplo es **netcat** o **nc**, escrita por Hobbit, (ver la siguiente Web Page http://www.atstake.com/research/tools/network_utilities) que se ha ganado el apodo "***TCP/IP Swiss Army Knife***", ya que puede cumplir una gran variedad de funciones. Hablaremos de ella en una futura edición.

Port Scanning – Contramedidas

El principal método de detección de un *Port Scanning* es utilizar un ***HIDS*** (*Host-based Intrusion Detection System*-Sistema de Detección de Intrusos basado en Sistemas Individuales). También es posible utilizar un ***NIDS*** con su placa de red configurada en modo promiscuo.

De la misma manera que la prevención ayudaba a evitar los *Ping Sweeps*, la correcta configuración y mantenimiento de los *routers* y/ó *firewalls* hará que sea más difícil que un intruso conozca los puertos/servicios/aplicaciones abiertos en los sistemas que se estén asegurando.

Detección de SO

Si prestamos atención a las respuestas que dio **nmap** al *TCP SYN scan*, podemos interpretar esos datos y deducir, siguiendo algunas premisas conocidas, que la máquina "víctima" tiene alguna clase de sistema operativo Windows (debido a los puertos 135 y 139 abiertos). Pero muchas veces, los puertos abiertos en un sistema no son fáciles de deducir y producen incertidumbre.

Aquí entra en juego nuevamente **nmap** que nos permite mediante el modificador **–O** identificar de acuerdo al *fingerprint* del *stack* **TCP** cuál es el sistema operativo de la "víctima". Vemos en la Figura 4 un ejemplo de detección de sistema operativo.

También existen otras herramientas disponibles para la detección de sistemas operativos, un ejemplo muy conocido es **QueSo** (www.apostols.org/projectz/). Es importante recordar que **QueSo** no es un *Port Scanner*, solamente hace detección de sistema operativo a través del puerto **TCP:80**.

Detección de SO – Contramedidas

Debido a que el proceso de detección de Sistema Operativo de una máquina "víctima" es esencialmente un análisis del fingerprint del stack TCP, las herramientas para evitar ésta técnica son las mismas que para evitar un *Port Scanning*: ***HIDSs*** y ***NIDSs***.

Descubrimiento automático

Existen herramientas muy completas diseñadas para englobar varias funcionalidades en el scanning. Mencionaremos dos: i) Cheops (**http://www.marko.net/cheops/**) que engloba usando una interfaz gráfica a ping, traceroute, port scanning, y scanning de SOs. ii) Tkined (parte del paquete Scotty, http://wwwhome.cs.utwente.nl/~schoenw/scotty/)

Conclusión

Hemos visto hasta aquí las principales herramientas y técnicas de *scanning* existentes, la información que es posible obtener y la utilidad de dicha información. Como postre a nuestro banquete de herramientas y técnicas, sólo queda nombrar una herramienta que permite hacer todas estas tareas en conjunto y desde una única interfaz: **Nessus** (www.nessus.org), una herramienta que consta de 2 partes, el Server está disponible sólo para plataformas Unix y/o Linux y el cliente está disponible para cualquier plataforma. El cliente funciona en modo gráfico.

En la próxima edición del Paso a Paso de Ethical Hacking veremos herramientas IDS, principalmente **Snort** y **Portsentry**.

Raúl Kuzner Analista en Sistemas

Three Way Handshaking

En toda comunicación TCP/IP, así como en la vida real, existen 2 (dos) interlocutores, de aquí en más llamaremos "*cliente*" al que inicia la comunicación y "*servidor*" a quien recibe y contesta la comunicación.

Cuando un cliente necesita comunicarse con un servidor, lo hace con algún servicio que está en ejecución en ese servidor. Así el cliente no sólo debe conocer la dirección IP del servidor, sino que también debe conocer el puerto donde este servicio está "escuchando".

En todo paquete TCP/IP, en el *header* (encabezado) del paquete, están la dirección IP y puerto de origen, la dirección IP y puerto de destino y un *flag* (bandera de estado) que establece el tipo de paquete. El estado de este *flag* tiene como propósito establecer un *three way handshaking* (saludo de 3 vías) para luego dar paso a un intercambio fluido de paquetes.

En la siguiente figura podemos ver un esquemático de este proceso de *Three-Way Handshaking*.

Cliente — Servidor
1) Envía Solicitud de Sincronización (SYN)
2) Responde Acordando la Sincronizacion (SYN/ACK)
3) Comienza el envió de paquetes con Payload (ACK)
4) | N-1) Continúa el intercambio de paquetes con Payload (ACK)
N) Envía Solicitud de Fin de Comunicación (FIN)
N+1) Responde Aceptando el Fin de Comunicación (FIN/ACK)
N+2) Responde Notificando (ACK)

Figura A – TCP/IP ***Three-Way Handshaking*** (Saludo de Tres Vías)

Claramente puede verse que el intercambio de paquetes para acordar la finalización de la comunicación responde al mismo criterio de *Tree-Way Handshaking*.

Para información más detallada sobre este proceso, vea el artículo "Entendiendo TCP/IP" publicado en la edición Nº 6 de NEX.

Rainbow Crack

Herramienta para cracking de passwords en Windows.

En este art culo detallamos la evoluci n de las herramientas para crackear los passwords de los sistemas operativos Windows. LopthCrack fue hasta hace muy poco indiscutida hasta la aparici n de RainbowCrack . Para poder comprender este art culo es muy importante conocer las tecnolog as de c mo se autentica (un usuario) bajo los sistemas operativos Windows y c mo se almacenan sus passwords. Aqu lo repasaremos brevemente.

Autenticación y las passwords

El proceso de presentar un usuario sus credenciales al momento del **logon** se denomina **autenticación**. Usualmente se realiza dando el userID (nombre de usuario) y password asociado. Aunque hoy, comienzan a ser populares otros métodos llamados en forma genérica **biométricos** (por huellas digitales, cara, voz, retina entre otros).

Existen muchas instancias en la que uno debe autenticarse en una red. Entre otras:

i) un logon a la red de nuestra empresa
ii) cuando accedo remotamente a la red de la empresa (dial-up o mediante una VPN)
iii) Acceso a un web-server en nuestra intranet o desde internet.
iv) Acceso wireless a un access point.

Cada uno de ellos tiene sus métodos y protocolos de autenticación.

Las **passwords** de los usuarios componen uno de los riesgos más grandes a la seguridad de las redes. Este riesgo incluye: la creación de las passwords, el modo en que los usuarios las protegen, cómo el sistema operativo las guarda y como las password son transmitidas a través de la red.

El sistema operativo es el responsable de guardar y transmitir a través de la red las "credenciales" (nombre de usuario y password) para las cuentas.

Windows 2000/2003 y XP soportan una variedad de distintos protocolos ***para transmitir*** las credenciales . También existen una variedad de formas de ***guardar las credenciales***.

Protocolos de autenticación de acceso a la red bajo sistemas operativos Windows

Los siguientes protocolos son soportados por Windows NT:

LAN Manager (LM)
NTLM
NTLMv2

Windows 2000-2003 y XP usan:

Kerberos v5

como el método de autenticación por default si utilizan Active Directory (AD). Ya que es muy posible que en nuestra infrestructura tengamos clientes "legacy" (Windows 95,98 etc) NT, Windows 2000/2003 y XP también soportan las autenticaciones anteriores (LM, NTLM y NTLMv2). Hay que recordar que éstas son autenticaciones más débiles que Kerberos y por lo tanto mucho más sencillas de comprometer.

Historia de Lopht crack

Hace unos años The Lopht mostró la debilidad de la autenticación LM de Windows. Lopht introdujo su programa de crackeo de passwords. Y, se transformó en el programa más popular de password-cracking del sistema operativo Windows. Lophtcrack LC5 (ver atstake INC., www.atstake.com), es una herramienta administrativa muy respetada y LM ha sido reemplazada por NTLM, NTLMv2 y Kerberos v5. Pero, hoy LC5 ha sido superado por Rainbow Crack.

> ***Ingeniería social:***
>
> A menudo la manera más sencilla de obtener información sobre una red o realizar una intrusión es preguntar. Aunque parezca raro, muchas veces los empleados queriendo o sin querer revelan información importante acerca de su empresa. Para el atacante, significa hacer la pregunta correcta a la persona indicada con el tono correcto. Esta explotación de confianza se conoce como: Ingeniería Social.

Importancia de conocer LC5 y Rainbow crack

Del uso de LC5 los administradores pudieron aprender cómo proteger aún más sus sistemas: promoviendo el uso de passwords complejos y sabiendo cómo proteger las cuentas importantes. Investigar y aprender cómo funciona Rainbowcrak es también una muy buena idea.

Sobre Rainbow crack

Utilizando un método llamado Master-Time memory Trade-Off Technique (basada en trabajos anteriores de Hellman), Phillipe Oechslin propuso una metodología capaz de crackear los passwors de Windows LM 12 veces más veloz que LC5. La idea es muy simple y se basa en usar tablas pre-calculadas con los hashes de todas las combinaciones posibles de caracteres en los passwords de Windows. Esto, junto a un algoritmo de búsqueda muy eficiente logra el factor 12 antes mencionado. Su trabajo original puede verse en el siguiente link http://lasecwww.epfl.ch/php_code/publications/search.php?ref=Oech03.

El método fue originalmente introducido para el protocolo LM, pero hoy es posible aplicarlo a NTLMv2 que permite utilizar el conjunto de caracteres Unicode (con mayúsculas y minúsculas) y usar hasta 128 caracters (LM permitía solo 14 y mayúsculas)

Al algoritmo se lo llama "Rainbow Crack".

Porqúe el factor 12 de Rainbow Crack es tan importante

El factor tiempo ha sido siempre considerado el más importante en password-cracking: si los passwords son suficientemente complejos, lleva mucho tiempo poder crackearlas y las hace seguras. Hoy máquinas mucho más poderosas, maquinas que pueden actuar en conjunto y ahora Rainbow Crack contribuyen a reducir los tiempos de crackeo.

Sin embargo existen una serie de acciones que podemos hacer para proteger nuestros sistemas de password cracking.



1. Utilizar protocolos de autenticación fuertes.
2. Utilizar protocolos de acceso remoto fuertes.
3. Utilizar comunicaciones seguras.
4. Proteger las bases de datos de las passwords.
5. Investigar las técnicas de password cracking.
6. Forzar políticas de passwords y realizar entrenamientos de concientización y auditorías de passwords.

> ### *¿Qué es SMB y SMB-signing?*
>
> Server Message Block (SMB) es un protocolo nativo soportado por todas las versiones de Windows. Aunque es básicamente un protocolo para compartir archivos, es también usado con otros propósitos. Uno de los más importantes: diseminar la información de Group Policy desde los Domain Controlers (DC) a los sistemas recién logueados. Desde el comienzo de Windows 2000 es posible mejorar la integridad de las sesiones SMB firmando digitalmente todos los paquetes en una sesión. Windows 2000-03 y XP pueden ser configurados para siempre firmar, nunca firmar o solo firmar si el otro sistema lo solicita.

1. Utilizar protocolos de autenticación fuertes.

Kerberos es el protocolo de autenticación de member servers o workstations en dominios con Windows 2000/2003/XP. En caso que tengamos sistemas "legacy" habrá que configurar que sea NTLMv2 (y no LM o NTLM) el protocolo de acceso. Esto puede hacerse:

>> Con group policies, (en dominios Windows 2000/2003).
>> En Windows NT SP4, modificando el Registry.
>> En Windows 95/98, instalando "Active Directory" client y modificando el Registry.

Es importante eliminar el almacenamiento de los hashes LM en la base de datos de las passwords. Esto está por default en Windows 2003. Se debe modificar el Registry para sistemas operativos anteriores.

2. Utilizar protocolos de acceso remoto fuertes

Usuarios en su casa o empleados fuera de las oficinas accederán remotamente a nuestra red. En muchos casos es importante imponer autenticación. El acceso remoto puede ser vía dial-up (telefónico o VPN), WEB o wireless.

Existen en Windows un número muy grande de metodologías de autenticación cuando se accede remotamente: incluyendo anonymous, basic (passwords en texto plano), integrated (variantes de LM o Kerberos), PAP, CHAP, MS-CHAP, MS-CHAPv2 and EAP (y variantes como PEAP y smart cards).

Los seteos por default (por defecto) son en general bastante débiles y es preciso llevarlos a lo máximo posible. Recordemos que encima de los problemas de autenticación en nuestra red aquí estamos realizando la comunicación en redes no confiables.

En los entornos de Windows 2000/2003 es posible definir además políticas de acceso remoto (Remote access policies) que permite la configuración de opciones para la autenticación en el acceso remoto mucho más granular (por ejemplo definir pertenencia a ciertos grupos, en determinado horario, tipo de protocolo y otros). Cuando sea apropiado convendrá utili-

> ### *LC5 : LophtCrack 5 (de @stake Inc. www.atstake.com)*
>
> **@stake LC 5** es la última versión de la herramienta más prestigiosa de auditoria y recuperación de passwords. Ayuda a los administradores auditar en forma completa y recuperar passwords de cuentas de usuarios y administradores del mundo Windows y Unix. De este modo se puede acceder a passwords perdidas o canalizar la migración a otro sistema de autenticación diferente.

zar IAS (Internet Authentication Server , RADIUS)

3. Utilizar comunicaciones seguras

Si las credenciales viajan, debemos proteger las comunicaciones. Estos métodos incluyen:

- VPNs
- SSL
- IPsec
- SMB signing

> ***Técnicas de cracking de passwords***
>
> **1. Ataque de diccionario**
> Utiliza un archivo que contiene palabras del diccionario. Usando el mismo algoritmo que usa Windows para crear el hash de los passwords, los compara con los hashes que el sistema operativo Windows tiene guardados.
>
> **2. Ataque heurístico**
> Se utiliza el conocimiento de cómo la mayor parte de los usuarios construyen sus passwords (por ejemplo agregado de números al final de palabras etc.) como ayuda para ir generando los passwords para comparar.
>
> **3. Ataque Brute Force (Fuerza Bruta).**
> Simplemente prueba todas las combinaciones posibles de caracteres.

4. Proteger las bases de datos de las passwords

Todo sistema operativo, por razones de seguridad contendrá uno o más archivos con una base de datos con la información de los usuarios. Esta base de datos alojará toda la información del usuario, incluyendo su password. Por supuesto que esos archivos contendrán la información, al menos los passwords, encriptados de modo de impedir su conocimiento en caso de ser comprometido el archivo. Por eso, resulta imperativo proteger esos archivos.

En Widows NT esa información se guardaba en la SAM (Security Accounts Manager) database. Hoy toda la familia de SOs Windows para workstations incluyendo Windows 2000 Professional y XP también contienen y usan una SAM. Por default Windows 2003 Server también contiene y usa una SAM. Pero, cuando organizamos nuestra infraestructura en dominios, con Active Directory (AD), los llamados "Domain Controlers (DC) contendrán la base de datos centralizada debidamente protegida en el archivo NTDS.DIT. Por ejemplo este archivo no puede copiarse si el sistema está activo.

Es por esto que debe tener en cuenta los siguientes consejos:

>> Proteja los backups. Los backups contienen copias de la SAM o ntds.dit. No los deje sin protección y monitoree su acceso.

>> Proteja físicamente las computadoras. Si el sistema puede accederse físicamente, un atacante puede ser capaz de rebootearla en otro sistema operativo. Esto le permitirá copiar el archivo de la base de datos para llevárselo o quizás realizar un ataque localmente.

>> Las cuentas de administrador deben ser limitadas, muy bien asignadas y auditadas. Existen herramientas administrativas como ERD Commander y variantes de pwdump que en manos de un atacante con privilegios de administrador permiten crackear las passwords del sistema.

5. Investigar las técnicas de password cracking

Un programa de password cracking será una excelente inversión para el grupo de seguridad de la empresa. Conocer las herramientas que un posible intruso usará nos preparará para la defensa. Tendremos noción de las posibles metodologías y más importante aún, de los tiempos que puede llevar realizar un cracking de passwords en nuestros sistemas.

LC5 de atstake Inc. ha sido desde hace años la herramienta para permitirnos entender y realizar auditorias de passwords. LC5, también utiliza tablas de RainbowCrack. Ver más detalles en recuadro adjunto. Si uno desea testear las tablas de RainbowCrack directamente, existen gran número de web-sites con información detallada.

6. Forzar políticas de passwords y realizar entrenamientos de concientización y auditorías de passwords

Implemente políticas de passwords. Por ejemplo, si utiliza passwords con más de 15 caracteres automáticamente requerirá el uso del protocolo de autenticación NTLM. Recordar que cuanto más larga es la password más difícil será crackearla. No solo el largo del password es importante sino la utilización de caracteres menos comunes dificultarán el crackeado.

Para proteger nuestros sistemas no basta con implementar políticas y tecnologías. Es fundamental promover entrenamiento de concientización a nivel de los usuarios. Es difícil pretender que el usuario entienda la importancia de seguir políticas de passwords. La concientización reducirá por tanto el esfuerzo requerido para lograr la aplicación de políticas.

> ***Si alguien puede acceder físicamente (y sin restricciones) a su computadora, dejó de ser suya.***
>
> Una vez que el atacante tiene acceso físico a una computadora es poco lo que se puede hacer para que él no logre tener privilegios de administrador sobre el Sistema Operativo. Con la cuenta administrador comprometida casi todos los datos que están permanentemente guardados pueden ser accedidos. También podría este atacante introducir hardware o software para monitorear keystrokes sin que el usuario se entere. Si una computadora ha sido comprometida físicamente o UD está en duda, deje de confiar en esa máquina.

Se deben realizar esfuerzos de enseñar como crear passwords complejas y demostraciones de cómo se pueden crackear password. Cuando la gente toma conciencia de cuan fácilmente se pueden crackear password muy sencillas los mentaliza al uso de passwords más complejas. También es posible entrenar de cómo impedir social engineering (ver recuadro adjunto).

Referencias:

Microsoft Windows Security Resource Kit por Microsoft Corporation, Ben Smith, Brian Komar, Elliot Lewis, y miembros de MS security Team. Microsoft Press.

> ***¿Cuándo debo usar tablas pre-computadas de hashes de passwords?***
>
> Tablas pre-computadas de passwords incluyen trillones de hashes de passwords que han sido computadas de antemano al proceso de auditoría de passwords y procesos de recuperación. La gran ventaja de estas tablas está en la reducción del tiempo requerido para recuperar una password individual.
>
> Durante la auditoría o recuperación, el hash de cada cuenta es comparado con los hashes en la tabla pre-computada. Que un hash coincida significa que la password ha sido recuperada. Este proceso permite reducir de forma drástica el tiempo requerido. Una sola cuenta puede llevar horas con un ataque tipo fuerza bruta y sólo segundos usando las tablas pre-calculadas. Sin embargo, los tiempos ahorrados usando tablas pre-computadas se reducen cuando el número de cuentas auditadas o recuperadas aumenta. Se recomienda usar tablas pre-computadas cuando el número de cuentas es menor de 2500. Cuando el número es mayor a 2500 se recomienda un análisis tipo fuerza bruta.

Modelos de Negocio basados en Software Libre

Es muy importante lograr entender las relaciones entre el software libre y la empresa. En este art culo se pretende ofrecer una visi n del software libre desde el punto de vista del emprendedor. Se van a recorrer algunos modelos de negocio relacionados con el software libre con viabilidad comprobada en el mundo real. Sabemos que existen multitud de modelos y variantes que no vamos a tratar ya que posiblemente exceder a el objetivo del art culo.

1. Introducción

El software libre ha experimentado un crecimiento muy importante en la última década. Al principio apoyado por personas que vendían ideas, como R.M.Stallman o E.S.Raymond, construyendo una percepción del software libre alrededor del concepto de «Hacker» como programador excepcional motivado por la calidad de su código.

Se hablaba entonces de las posibilidades de Linux y otros proyectos de software libre de perdurar en el tiempo y de convertirse en alternativas reales de propósito general. Era necesario alcanzar una determinada masa crítica de usuarios que garantizara la supervivencia del modelo y su crecimiento.

Casi desde el principio aparecieron intentos de poner en marcha negocios alrededor del software libre aunque por regla general no tuvieron excesivo éxito (honrosa excepción de algunas empresas como RedHat). Se cuestionaba bastante la viabilidad de negocios alrededor del software libre, tras fracasos importantes de empresas como VALinux o Corel.

En los últimos tres o cuatro años el concepto de Software Libre asociado a la imagen de «Hacker» ha cambiado radicalmente. Se ha producido una revolución muy importante que consiste en la adopción de grandes proyectos de Software Libre (en especial Linux, Apache y MySQL) por parte del entorno empresarial. Esto ha supuesto para el Software Libre la capacidad de ofrecer parte de las necesidades que el concepto «Hacker» no podía proporcionar y que el entorno empresarial demandaba, como soporte profesional, acuerdos de nivel de servicio, compatibilidad con otras plataformas o certificaciones en hardware y software.

Hoy en día ya no se habla de la masa crítica de usuarios, ni de la viabilidad empresarial del software libre. Este nivel se ha alcanzado ya. En este momento existen multitud de modelos de negocio alrededor del software libre con viabilidad demostrada y probada. En este artículo se pretende ofrecer una visión del software libre desde el punto de vista del emprendedor. Se van a recorrer algunos modelos de negocio relacionados con el software libre con viabilidad comprobada en el mundo real. Sabemos que existen multitud de modelos y variantes que no vamos a tratar ya que posiblemente excedería el objetivo del artículo.

A medida que el Software Libre vaya creciendo en número de usuarios y empresas que lo utilizan irán apareciendo necesariamente nuevos modelos de negocio no recogidos en este artículo. El presente artículo sirvió como prólogo a la ponencia que se presentó en el ultimo congreso hispaLinux en Septiembre de 2003, cubriendo el análisis de los diferentes modelos de negocio viables dentro del mundo del Software Libre, considerando como marco de referencia el avance experimentado por estas tecnologías en los últimos años y desde una óptica de comparación con el software distribuido bajo licencia propietaria, ofreciendo un imagen presente y una valoración futura de la viabilidad de estos modelos de negocio.

2. Cambio de planteamiento: de producto a servicio

Desde el punto de vista empresarial, el software libre supone un cambio de planteamiento con respecto al software propietario tradicional. El cambio fundamental consiste en el paso de obtención de ingresos por venta de productos a obtención de ingresos por venta de servicios.

En general, el concepto de software libre invalida la obtención de ingresos por repetición de ventas de licencias de uso de un mismo producto cerrado, ya que, por su propia naturaleza, cualquier persona que obtenga ese software es capaz de modificarlo y copiarlo dentro de las restricciones impuestas por cada licencia. Ninguna empresa pagaría por un producto (recalco lo de producto) si es capaz de obtenerlo gratuitamente a no ser que el desembolso económico suponga una serie de servicios adicionales por los cuales sí que estaría dispuesta a pagar.

Mientras que el software propietario basa sus ingresos en las licencias de uso de software, el software libre tiene que buscar su rentabilidad en los servicios asociados a ese software y no en el producto en sí.

Uno de los casos más brillantes podemos encontrarlo en la empresa MySQL AB. Esta empresa ha construido su modelo de negocio alrededor de un software de Base de Datos (MySQL) el cual, en muchos aspectos (escalabilidad, soporte al cliente final, características técnicas, etc.), está por detrás de grandes productos comerciales como Oracle o DB2.

Las posibilidades de sobrevivir de MySQL AB compitiendo con estos otros productos son nulas en un ámbito puramente comercial. Pero MySQL es software libre, cualquiera puede instalarlo y utilizarlo, y cubre todas las necesidades de la mayoría de las implantaciones. Esto ha hecho que MySQL se encuentre instalado en más de cuatro millones de sistemas en producción (no es posible conocer el número exacto, ya que nadie vende licencias de uso y es posible la descarga y uso del software casi sin limitaciones). Es impensable que MySQL hubiera logrado esos números compitiendo con Oracle o DB2.

MySQL AB renuncia a parte de los ingresos obtenidos a través de licencias a cambio de incrementar la base de usuarios y obtener ingresos entre otras áreas, como los contratos de soporte y consultoría o las peticiones de personalización que algunos de estos usuarios solicitan a la empresa. Las mejoras logradas en el software son incorporadas al propio producto de modo que pasan a estar disponibles para todo el mundo en la siguiente versión.

Debido a este proceso, la base de datos MySQL está inmersa en un desarrollo muy rápido y está alcanzando en funcionalidad a sus hermanos mayores. Los usuarios empiezan a demandar (e implementar) funcionalidades que solamente existían en grandes productos propietarios (escalabilidad, seguridad, etc.).

El uso de software libre todavía tiene una serie de resistencias que, en algunas empresas, son casi insalvables hoy en día. En las pequeñas empresas todavía existe un cierto temor a los cambios y a la falta de compatibilidad con lo que ha existido hasta ahora. Es por ello que la adopción de software libre está siendo asumida primero por las grandes empresas y las entidades gubernamentales.

3. Ventajas competitivas

La utilización de software libre en el entorno empresarial ofrece una serie de ventajas competitivas muy importantes, pero que suponen un cambio de mentalidad sobre un modelo (el clásico de venta de licencias) asumido completamente por el entorno. Algunas de las principales ventajas competitivas que puede ofrecer la utilización de software libre son las siguientes:

>>Capacidad de modificación del código: Con software libre una empresa tiene capacidad para adaptar la solución a sus necesidades, arreglar fallos operativos o de seguridad, etc.
>>Independencia del proveedor: La implantación de una solución basada en software libre permite al cliente la elección del mejor proveedor de servicios. La disponibilidad del código fuente y la capacidad para modificarlo permite que una empresa no quede atada a un determinado proveedor.
>>Seguridad: Se disminuye o incluso se puede llegar a eliminar la existencia de puertas traseras, troyanos, etc., debido a que cualquier empresa puede auditar el código fuente de las aplicaciones que pone en producción. Adicionalmente la propia «comunidad de usuarios» está desarrollando constantemente el mismo trabajo de auditoria de código.
>>Garantías de permanencia: La utilización sistemática de estándares hacen difícil que una determinada aplicación pueda quedar sin soporte. En el caso de que esto suceda, la disponibilidad del código permite que otro grupo de usuarios u otra empresa pueda tomar el liderazgo en el desarrollo. Las aplicaciones basadas en software libre solamente mueren cuando dejan de utilizarse, generalmente porque aparezcan aplicaciones superiores que actúen como sustitutivas.

Desde el punto de vista del empresario, el desarrollar software libre también tiene una serie de ventajas indudables, como son:

>>Disponibilidad de una comunidad potencial enorme de programadores y probadores del software
>>Posibilidad de respuesta rápida ante clientes por problemas en el código, consecuencia de la anterior
>>Evolución «automática» de las distintas piezas de software que componen una solución
>>Tendencia a una calidad enorme en el código desarrollado, como consecuencia de la necesidad de trabajar en equipo con personas desconocidas
>>Tendencia a la utilización de estándares, lo que permite la construcción de soluciones mucho más completas mediante integración de distintas aplicaciones

Es difícil para una empresa que invierte recursos en el desarrollo de software la decisión de liberar el código. El éxito de la iniciativa depende en gran medida de la utilidad real que el software en cuestión tenga para la mayoría de los usuarios.

4. Algunos modelos de negocio basados en software libre

En esta sección vamos a comentar una serie de modelos de negocio relacionados con software libre. Evidentemente no están cubiertos todos ellos, ya que casi cada día aparecen nuevas ideas o planteamientos, algunos de ellos muy exitosos. Hemos dejado fuera conscientemente algunos modelos de negocio perfectamente válidos y exitosos como los relacionados con formación o consultoría ya que son esencialmente iguales si están basados en software libre o en software con licencia propietaria. No todos los modelos de negocio comentados se basan en software libre puro. Algunos de los modelos de negocio relacionados con software libre más exitosos se basan en la mezcla en mayor o menor medida de software libre y software propietario. Hay un concepto clave para comprobar la viabilidad de un negocio. Los modelos de negocio basados en software libre más exitosos se logran cuando los dos actores principales detrás del desarrollo de una determinada tecnología (el propietario del software y la «comunidad») se encuentran motivados para mantener y hacer evolucionar dicha tecnología (léase más adelante el ejemplo de TrollTech con las librerías QT).

4.1 Software libre como plataforma a software comercial

No se puede decir que este sea un modelo de negocio basado en software libre propiamente dicho, pero ciertamente tiene bastante interés para completar el «gran dibujo» que supone la relación entre el mundo del software libre y el mundo del software propietario.

Algunas grandes empresas comerciales tradicionales como podría ser Oracle, o en su momento Corel, han seguido con cautela la adaptación al software libre. Con piezas de código cerradas que suponen parte de su «core-business», han intentado (con mayor o menor éxito) la integración con el mundo del software libre tratando de mejorar la compatibilidad y permitiendo la ejecución de su software sobre sistemas operativos libres (como Linux).

Este modelo de negocio consiste básicamente en la adaptación de determinadas aplicaciones disponibles bajo licencia propietaria de modo que pueda coexistir con aplicaciones basadas en software libre. Hoy en día existen multitud de empresas siguiendo este modelo de negocio.

Algunos ámbitos de necesidad todavía no están cubiertos por software libre, aunque rápidamente se están llenando todos los huecos. Muchas empresas cubren estas necesidades desarrollando software propietario capaz de ser ejecutado en sistemas libres como Linux.

Hay multitud de ejemplos que podemos citar, como:

>> SAP/R3
>> Siebel
>> Macromedia Flash
>> HP openview

Todos ellos están cubriendo necesidades existentes en el mundo del software libre y que no se encuentran todavía cubiertas satisfactoriamente por aplicaciones libres. Este modelo de negocio no es válido si asumimos como tal la capacidad de obtener ingresos sostenidos de un determinado esfuerzo. Existen históricamente multitud de ejemplos en el ámbito del software libre que lo corrobora, como por ejemplo:

>>ApplixWare o Corel WordPerfect Office, son suites ofimáticas propietarias que pretendían cubrir las necesidades ofimáticas en Linux. Hoy en día prácticamente han desaparecido con el crecimiento de aplicaciones libres sustitutivas.

>>SSH, protocolo de comunicaciones propietario, desaparecido rápidamente tras la aparición de openSSH.

>>IPlanet Enterprise Server, es un servidor web con licencia propietaria prácticamente en extinción por el éxito de Apache.

Hay algunas aplicaciones que pueden encontrarse en un callejón sin salida si no logran encontrar un nicho de usuarios suficientemente importante en el mundo del software libre, como pueden ser los diversos servidores de aplicaciones, por el éxito de alternativas libres como Tomcat, JBoss o Jonas, sistemas de Firewall propietarios como FW1 o servidores de Directorio LDAP como IPlanet Directory Server.

Este modelo de negocio planteado no logra captar las ventajas competitivas derivadas de modelos de negocio puramente basados en software libre por lo que pueden llegar a tener dificultades para obtener rentabilidad a largo plazo. Si la necesidad existe y es real, a la larga aparecerán proyectos de software libre que cubran dicha necesidad, con todas las ventajas inherentes que proporciona el uso de software libre.

4.2 Modelo de desarrollo con doble licencia

Este es uno de los nuevos modelos de negocio que se ha comprobado que son exitosos. Básicamente se trata de asociar dos licencias a un determinado código, una de ellas es licencia libre y otra es licencia propietaria. Como usuario, puedes elegir cualquiera de las dos licencias para aplicar al uso del software. Se podría asumir que este modelo de negocio está llegando a la madurez debido a que ha sido probado durante bastantes años con éxito por múltiples empresas.

Ejemplos: Trolltech y las librerías QT , y openOffice

4.3 Soporte y productos alternativos

Este modelo es bastante tradicional y consiste en desarrollar una determinada aplicación bajo licencia libre, ofreciendo personalizaciones y/o servicios específicos sobre este software. Como consecuencia de este modelo de negocio generalmente se evoluciona a la disponibilidad de dos versiones del mismo software, una libre y otra propietaria, ofreciendo esta última una funcionalidad superior.

Ejemplos: Sendmail , MySQL

4.4 Desarrollo de componentes comerciales para productos de software libre

Este modelo es muy similar al anterior. Consiste en el desarrollo bajo licencia libre de aplicaciones específicas, generalmente de propósito general. Sobre estas aplicaciones se desarrollan determinados componentes comerciales que se distribuyen bajo el clásico modelo de licencias de uso que cubren determinadas necesidades específicas de un cliente.

Ejemplos: Evolution , Kivio

4.5 Donaciones o suscripciones

Algunos negocios, sobre todo las publicaciones digitales, basan su estructura de ingresos en las suscripciones de los usuarios. Este tipo de modelo de negocio por regla general está ligado a las revistas electrónicas, aunque también se observa en algunos proyectos de software libre.

Ejemplos: Linux Weekly News (lwn.net), Typo3, Compiere , Transgaming y WineX

4.6 Nuevos productos derivados de licencias tipo BSD

Existe una diferencia fundamental entre las dos grandes licencias libres, GPL y BSD. Mientras que con la licencia GPL todo el software derivado está obligado a llevar la misma licencia, con las licencias tipo BSD es posible re-licenciar el código derivado a cualquier tipo de licencia, incluidas licencias propietarias. Productos con licencia derivada de BSD podemos encontrar Apache, Wine o XFree.

Ejemplos: Crossover Office / Plugin MacOS X

4.7 Integración de software

La integración de componentes o elementos de software no es un concepto nuevo. Desde hace muchos años el concepto de reutilización de código ha sido crítico en el éxito de una empresa de desarrollo. Lo que ha hecho el software libre es llevar la integración de software a límites mucho más extensos. El software libre por su naturaleza ofrece casi todas las ventajas para costruir un negocio basado en integración de software. Las bases de este tipo de negocio consisten en un potente I+D, capaz de conocer y analizar la mayoría de las herramientas disponibles basadas en software libre. De este conocimiento es posible construir aplicaciones a la medida del cliente mediante la integración de los elementos que más se adaptan a sus necesidades. La clave para este tipo de negocios es el desarrollo de software de integración evitando en la medida de lo posible la modificación de las aplicaciones a integrar y que permita evolucionar los distintos elementos de la solución final con el mínimo impacto sobre la solución alcanzada. Para las soluciones construidas en base a integración de distintas herramientas es vital la liberación del código desarrollado, sobre todo si la integración ha requerido la adaptación de alguna de las herramientas utilizadas.

Ejemplos: openSistemas: integrador de soluciones

Sobre el autor

Fernando Monera (correo electrónico: fmonera@opensistemas.com), Ingeniero Informático, MBA en ICAI, trabaja en la actualidad como director general y socio fundador de openSistemas (Web Page: http://www.opensistemas.com), empresa focalizada en integración de soluciones basadas en software libre.
Su relación con el software libre comenzó en 1997 interesado fundamentalmente por el aspecto sociológico detrás del concepto de software libre, representado en gran medida en el artículo «La catedral y el Bazar» de Eric S. Raymond.
En 2001 fundó un portal de supercomputación de habla hispana denominado hispaCluster (Web Page: http://www.hispacluster.org) en pleno funcionamiento hoy en día y patrocinado por openSistemas.
Desde 1997 hasta hoy ha estado relacionado con el software libre tanto laboral como personalmente, involucrado en diferentes proyectos como TWIG (herramienta groupware basada en plataforma web) o Jensen (linux para plataformas Alpha), fomentando el uso de software libre en todos los ámbitos que le ha sido posible.

(www.opensistemas.com),

Conclusiones

Hasta hace unos pocos años no se había podido generalizar la viabilidad del software libre como modelo de negocio. Por regla general, las buenas ideas fallaban por falta de masa crítica de mercado, falta de soporte de grandes empresas u otros motivos. En los últimos años, la creciente competitividad, el aumento de madurez del mercado tecnológico, el propio avance del Software Libre y su probada eficiencia y calidad junto con una situación económicamente débil, han provocado el comienzo de la asunción de un cambio de paradigma en el desarrollo y distribución de software. Los clientes empiezan a no estar satisfechos con la adquisición de productos, sino que quieren una personalización, una adaptación a sus problemas reales. De esta forma muchas empresas han pasado de un modelo basado en venta de producto a otro basado en venta de servicios y soporte asociados. Es en este nuevo modelo donde el Software Libre es casi imbatible. El cambio que se está produciendo tiene además implicaciones de orden político que no escapan a los gobiernos y grandes empresas y que provocan movimientos muy importantes impensables hace muy poco tiempo. Por ejemplo en España estamos viendo cómo el uso del software libre se utiliza en los programas electorales como un arma de diferenciación política.

Utilizando soluciones basadas en software libre no es necesario estar constantemente reinventando la rueda. Si nos paramos a pensar un poco es un modelo muchísimo más lógico desde el punto de vista del desarrollo de software. Los nuevos esfuerzos parten de código ya existente y disponible garantizando que los nuevos esfuerzos no tienen que partir de cero. Desde mi punto de vista, consiste en la implementación del método científico al desarrollo de software. Posiblemente, la consolidación de los modelos de negocio basados en software libre estaba pendiente del alcance de la masa crítica de usuarios necesaria para convertir el planteamiento idealista basado en el concepto de «hacker» en planteamientos de negocio mucho más sólidos que podemos encontrar en la actualidad.

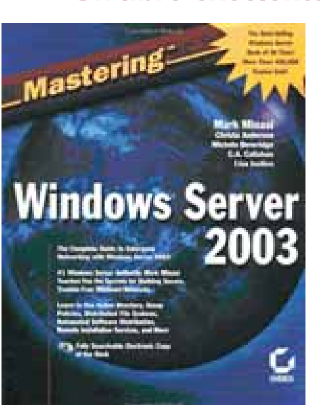
Fernando Monera Daroqui







Un libro excelente sobre Windows Server 2003.



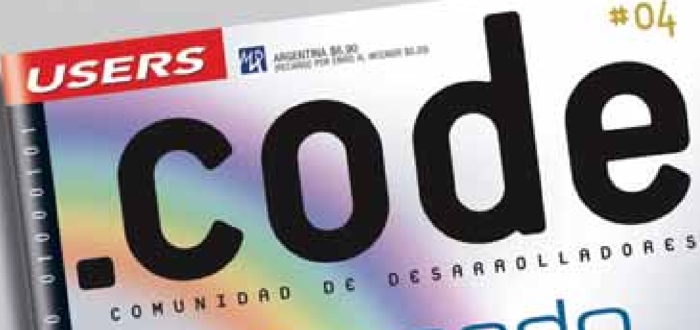
Mastering Windows Server 2003
by Mark Minasi (Editor), Christa Anderson, Michele Beveridge, C. A. Callahan, Lisa Justice

Pocas veces se puede recomendar tan abiertamente un libro. Este es el caso de Mastering Windows Server 2003 por Mark Minasi et al. (Sybex 2004).

Si usted es, o aspira a ser un Administrador o Consultor Windows, no busque más que "Mastering Windows Server 2003" de Sybex (en inglés). Aun ANAYA (quien sacó la traducción (hay que decir bastante pobre) al español del libro correspondiente a W2000) aún no lo ha editado. Su cobertura es profunda, comprensible, imparcial y altamente "legible" (algunos incluso dirían "entretenida").Su autor, Mark Minasi es una autoridad en el tema. Construido en la base de años de experiencia trabajando y escribiendo sobre productos Windows, Minasi lo lleva a conocer las tecnologías sobre las que se basa Windows 2003 Server (el sistema operativo de Microsoft que proporciona una solución para compartir archivos e impresoras, conectividad segura en Internet, el desarrollo de aplicaciones de escritorio centralizadas, y la colaboración entre negocios, empleados, y clientes).



#04



Mercado Laboral

¿Es lo mismo trabajar como desarrollador en cualquier lugar del mundo? ¿Cuánto podemos ganar? ¿Emigramos o nos quedamos en casa? ¿Programamos para el mercado interno o externo?



Las respuestas a éstas y otras preguntas, en un **INFORME ESPECIAL** del mercado de trabajo de los desarrolladores a nivel mundial. Los casos de la Argentina, México, Canadá, España y Asia.

ASP.NET 2.0 Las novedades de Whidbey para el desarrollo de aplicaciones móviles | XML: ejecución, optimización y compilación de consultas con XPath en .NET

C# ¿SI O NO? Analizamos este lenguaje desde la perspectiva de los programadores Java, C++ y Visual Basic | Compresión de archivos con UnRarLib en C++

DIRECTSOUND con C# | Cómo agregar sonido a las aplicaciones con este componente de Managed DirectX | Techniques: búsquedas utilizando árboles binarios

ADEMAS Reviews: MySQL 4.1 beta y 5.0 alpha - InstallShield X | Entrevista a Sanjay Sarathy, Senior Director of Sun Developers Network (SDN) | Correo de lectores

WHITE PAPER: UNIFIED MODELING LANGUAGE



News (!)

Anuncios para los Javeros

Conferencias gratuitas

Breves (!)

Ya está disponible Flash Lite 1.1

Un stop para los ataques DoS

Studio Express Beta

Microsoft Users Group - Paraguay



[especial]

mercado laboral



ESCALA JERÁRQUICA

[white paper]

UML: Unified Modeling Language

DirectSound con C#



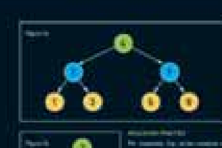
Los archivos WAV

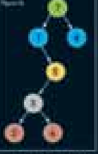
Propiedades de BufferDescription

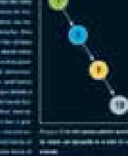
toolbox

ÁRBOLES BINARIOS DE BÚSQUEDA

Búsqueda con árboles B+







Otras consideraciones

Unix, BSD y Linux

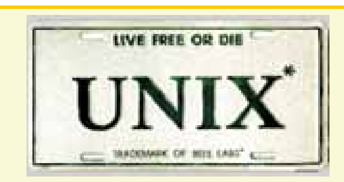
De UNIX a BSD

La historia del sistema UNIX data de los años 60, cuando los laboratorios Bell de AT&T y el fabricante de computadoras General Electric (GE) trabajaron sobre un sistema operativo experimental denominado MULTICS. MULTICS, de MULTiplexed Information and Computing System (Información multiplexada y sistema de computación) fue diseñado como sistema operativo interactivo para la computadora GE645, permitiendo compartir información al tiempo que proporcionaba seguridad. El desarrollo sufrió muchos retrasos, y las versiones de producción resultaron lentas y con grandes necesidades de memoria. Por una serie de razones, los Laboratorios Bell abandonaron el proyecto. Sin embargo, el sistema MULTICS implementó muchas características innovadoras y produjo un entorno de computación excelente.

Ken Thompson y Dennis Ritchie, los años 70



En 1969, Ken Thompson, uno de los investigadores de los Laboratorios Bell involucrado en el proyecto MULTICS, escribió un juego para la computadora GE denominado Space Travel. Este juego simulaba el sistema solar y una nave espacial. Thompson vio que el juego se ejecutaba a tirones sobre la máquina GE y resultaba muy costoso -aproximadamente 75 dólares por ejecución-. Con la ayuda de Dennis Ritchie, Thompson volvió a escribir el juego para ejecutarse sobre un DEC PDP-7. Esta experiencia inicial le dio la oportunidad de escribir un nuevo sistema operativo sobre el PDP-7, utilizando la estructura de un sistema de archivos que habían diseñado Thompson, Ritchie y Rudd Canaday. Thompson, Ritchie y sus colegas crearon un sistema operativo multitarea, incluyendo un sistema de archivos, un intérprete de órdenes y algunas utilidades para el PDP-7. Más tarde, una vez que el nuevo sistema operativo se estaba ejecutando, se revisó el Space Travel para ejecutarlo sobre él. Muchas cosas en el Sistema UNIX proceden de este simple sistema operativo.



Licencia repartida en eventos por Digital Equipment Corporation (DEC) en los años 80. La historia es muy interesante y puede leerse en el siguiente link: http://www.unix.org/license-plate.html

Puesto que el nuevo sistema operativo multitarea para el PDP-7 podía soportar dos usuarios simultáneamente, se le llamó humorísticamente UNICS de UNiplexed Information and Computing System (Información uniplexada y sistema de computación); el primer uso de este nombre se atribuye a Brian Kernighan. El nombre se cambió ligeramente a UNIX en 1970, y ha permanecido así desde entonces.

Ken Thomson se unió a Dennis Ritchie quien escribió el primer compilador C. En 1973 ellos rescribieron el Kernel de UNIX en C. Al siguiente año, una versión de UNIX conocida como Fith Edition (quinta Edición) fue licenciada las universidades. La Seventh-Edition (Séptima Edición), que aparece en 1978, sirvió como un punto de bifurcación para el desarrollo de dos líneas de UNIX. Estas son conocidas como SRV5 (System V) y BSD.

De BSD a FreeBSD, NetBSD y OPenBSD

La evolución de estas dos versiones de UNIX (System V y BSD), se realiza en forma muy entrelazada. Fines del 80 y comienzos del 90 vieron conflictos entre estas dos divisiones. Después de muchos años, cada variante había adoptado muchas de las características del otro. Desde el punto de vista comercial fue System V quien ganó logrando estandarizar gran parte del código. Y la mayor parte de los vendors adoptaron System V.

Sin embargo, System V tomó muchísimas modificaciones aportadas por BSD. En síntesis el resultado fue una fusión de ambas ramas. La rama BSD no murió, por el contrario fue muy utilizada para la investigación, implementada para hardware de PC y para servidores de un solo propósito (por ejemplo muchísimos web-servers

El Pasado de UNIX

"...el número de instalaciones UNIX ha crecido a 10, se esperan más… "

- Dennis Ritchie y Ken Thompson, Junio 1972

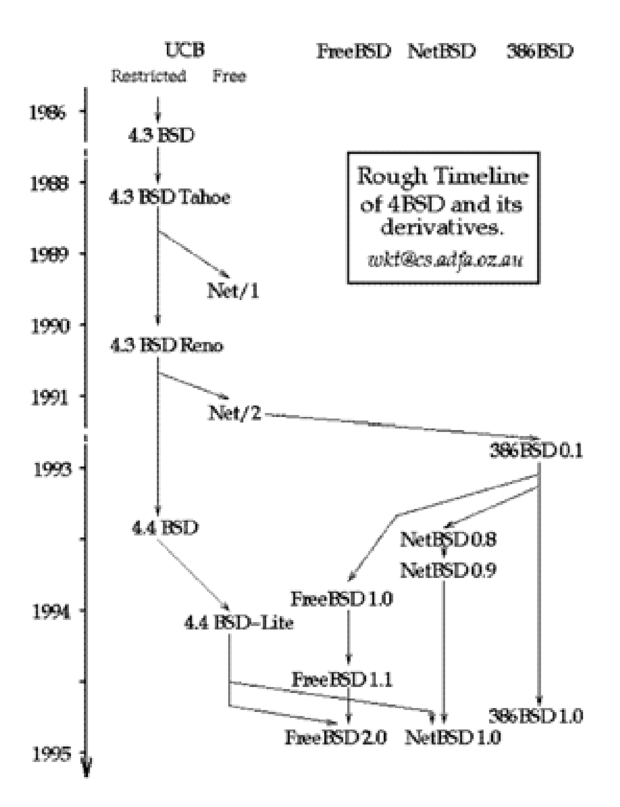
"... Cuando los laboratorios Bell se separaron del proyecto, ellos necesitaron escribir un sistema operativo de modo de poder continuar jugando a "guerra de las galaxias" en una máquina de menores recursos (una DEC PDP-7 [Programmed Data Processor] con 4K de memoria para programas de usuarios). El resultado fue un sistema que fue bautizado por un colega como UNICS (UNiplexed Information and Computing System) — un "MULTICS mutilado"; nadie recuerda de quien fué la idea de llamarlo UNIX "

The Unix Family Tree

UNICS (1969)
Fifth Edition (1973)
Sixth Edition (1976)
Seventh Edition (1978)

SVR5 (1983)
SunOS 5.x/Solaris (SUN)
HP-UX (HP)
AIX (IBM)
IRIX (SGI)
Digital UNIX (formerly OSF/1) (DEC)

BSD (1979)
SunOS 4.x (SUN)
ULTRIX (DEC)
NextStep (NeXT)



utilizan algún derivado de BSD).

El resultado fue la aparición de muchas versiones diferentes de Unix, todas basadas en seventh edition. Muchas de estas eran propietarias y mantenidas por sus respectivos marcas de harware (ejemplo Solaris de Sun es una variante de System V). Tres versiones de la rama BSD de Unix terminaron Open Source (código fuente abierta): FreeBSD (que se concentró en sencillez de instalación apuntando a la plataforma PC), NetBSD (se concentró en muchas diferentes arquitecturas), y una variante de NetBSD: OPenBSD (concentrada fundamentalmente en seguridad).

Referencias de Unix:

http://www.datametrics.com/tech/unix/uxhistry/brf-hist.htm,

¿Cómo nace BSD?

Durante un sabático realizado en la Universidad de California, Berkeley, Ken Thomson introdujo UNIX. Ya en 1978, estudiantes de Berkeley habían comenzado a producir versiones de Unix customizadas (BSD, Berkeley Software distribution). Durante los 80, Berkeley manejó un contrato con el departamento de defensa para incorporar TCP/IP en BSD y producir un sistema operativo estandarizado para las computadoras del departamento de defensa. Con la aparición del 4.3BSD y del llamado Berkeley Networking Release 2 tapes ("Net/2"), Berkeley había creado un sistema operativo completo, independiente del codigo de AT&T.

William Jolitz comenzó portando BSD a la plataforma 386, escribiendo una serie de artículos para la revista Dr. Dobb's journal. Este software fue llamado "386BSD". Para 1993, Jolitz había decidido detener el trabajo en una versión mejorada de 386BSD. Este fue el comienzo del BSD moderno en sus tres variantes: FreeBSD, netBSD y OpenBSD.

http://perso.wanadoo.fr/levenez/unix, y http://www.crackmonkey.org/unix.html.
Mucha más información sobre la historia de Unix se puede hallar en un trabajo de Kusick 1999 y
ftp://ftp.freebsd.org/pub/FreeBSD/FreeBSD-current/src/share/misc/bsd-family-tree.

Una Breve historia sobre el lenguaje C

El lenguaje de programación C, fue desarrollado en los Laboratorios Bell durante los primeros años de los 70. Surgió de un lenguaje de computadoras llamado B y de uno anterior BCPL. Inicialmente fue introducido para el desarrollo del sistema operativo (SO) UNIX. Pero, luego se expandió su uso a diferentes SOs. La versión original de C se conoció como C K&R por Kernighan and Ritchie autores del libro "The C Programming Language". A medida que el lenguaje se desarrollo y estandarizó, se hizo dominante una versión conocida como ANSI (ANSI (American National Standards Institute) C. Si estudia este lenguaje sepa que se encontrará con referencias como C K&R o ANSI C. Aunque ya no es el lenguaje más popular para nuevos desarrollos, aún se usa para algunos SOs, programación de protocolos de red y sistemas embebidos. Más aún, es mucho el software "legacy" programado en C y que debe ser mantenido

LINUX

Linux nace con la tesis de Maestría que desarrolla Linus Torvalds en Helsinki, Finlandia en 1991. Para ese momento

la Free Software Foundation (FSF) había desarrollado una serie de aplicaciones/librerías para Unix. Con ese material y otros componentes (en particular algunos componentes de BSD y del software X-Windows desarrollado en el MIT) se produjo un sistema operativo libre de ser modificado y muy útil. Es decir, se combinó el "Linux Kernel" de Linus Torvalds con lo aportado por el proyecto GNU, para crear el sistema operativo "Linux", también llamado GNU/Linux.

Dentro de la comunidad Linux, se combinaron de forma diferente componentes disponibles. Cada combinación se llaman "distribución". Ejemplos de organizaciones que realizan tales distribuciones son: Red Hat, Mandrake, SuSE, Caldera, Corel, y Debian.



Existen diferencias entre ellas, pero se basan en los mismos pilares: el Kernel Linux y las librerias glibc de GNU. Como ambas están cubiertas por licencias tipo "copyleft", los cambios son realizados por todas las distribuciones. Esto ha creado una fuerza unificadora que no existe entre BSD y los UNIX derivados de AT&T.

Hugo Cela - Consultor IT



Qué hace de OpenBSD el Sistema Operativo más seguro del mundo?

Sólo una vulnerabilidad ("remote hole") en la instalación por defecto en más de 8 años!



OpenBSD (ver web www.openBSD.org) es un sistema operativo desarrollado conforme a los estándares de Unix. Está basado en BSD 4.4, lo que implica más de 25 años de evolución continua y maduración. Pero es más conocido por la siguiente frase: "OpenBSD es el sistema operativo más seguro del mundo"

Gracias al proyecto OpenBSD podemos disponer de un sistema operativo similar a Unix, robusto, funcional, seguro y libre.

Basándose en el código de NetBSD, Theo de Raadt fundó el proyecto OpenBSD en 1996. Así comenzó con lo que sería la auditoria más grande del mundo, para hacer de OpenBSD el sistemas operativo más seguro que haya existido sobre el planeta. Luego de un estudio riguroso del código se lo optimizó para segurizarlo y se le integró criptografía. OpenBSD fue concebido para ser seguro desde su nacimiento.

Se jacta de no tener agujeros en su instalación por defecto (1 en 8 años!!) y de los pocos recursos de hardware necesarios para correr. Y con razón: durante años su instalación no tuvo vulnerabilidades remotas y requiere solamente una 486 con 16 MB de RAM para funcionar. Además, toma minutos instalarlo.

El desarrollo de OpenBSD se concentra en la corrección de código, la seguridad proactiva y la criptografía, características que lo transforman en un OS excelente para entornos donde la seguridad es vital. Por no estar atado a compromisos comerciales como los OSs propietarios pueden implementar características de seguridad de manera mucho más rápido. Además fue el primer OS que incluyó IPsec (desde la versión 2.1 en 1997), es prácticamente imposible generar un buffer overflow en él (por su sano código) e incorpora criptografía desde las raíces más hondas del sistema entre otras características de seguridad.

Recomendamos leer el artículo de Luciano Bello (lbello@sistemas.frba.utn.edu.ar) que está disponible en http://www.lucianobello.com.ar/a_secure_os.html

¿Qué es Minix?

Minix es un clon UNIX que está accesible con todos sus códigos fuente. Debido a su pequeño tamaño, un diseño basado en micro-kernel y amplia documentación, es muy apropiado para quienes quieran correr un sistema operativo tipo-UNIX en su computadora personal y aprender cómo funcionan estos sistemas operativos. Es posible para alguien que no está familiarizado con detalles de sistemas operativos poder comprender casi todo el sistema en pocos meses de uso y estudio. MINIX fue escrito desde cero y no contiene ningún código de AT&T. Ni en el kernel, compilador, utilidades o librerías. Por esta razón las fuentes completas están disponibles vía ftp o WWW.

Minix fue escrito alrededor de 1987 por Andrew Tanenbaum, académico de la Vrije Universiteit, Amsterdam, Holanda. Referimos al lector al excelente libro de Tanenbaum: *Operating Systems: Design and Implementation*, ISBN 0-13-637331-3.

Una versión reducida en 12.000 líneas de código, mayormente escritas en C (del kernel, administrador de la memoria y el file system) están contenidas en este libro.

El desarrollo de linux estuvo influenciado por Minix. Al momento de su desarrollo por Linus Torvalds, la licencia de Minix era considerada muy liberal, con un costo de licencia muy reducido (casi formal) en comparación con otros sistemas operativos competencia. Sin embargo, debido a no ser completamente Open Source, esfuerzos de desarrollo se volcaron a los Kernels de Linux y FreeBSD. A fines de los 90, la licencia de Minix fue convertida a open source, pero ya eran muy pocos los involucrados en su desarrollo.

Recomendamos ver el sitio WEB de Andrew S Tanembaum
http://www.cs.vu.nl/~ast/minix.html

Agosto de 1991

"Hello everybody out there using Minix"... estoy escribiendo un sistema operativo (gratis), (solo es un hobby, no va a ser grande y profesional como GNU), para clones AT 386 y 486. Se ha estado cocinando desde abril y está empezando a quedar terminado. Me gustaría recibir algún feedback sobre cosas que a la gente le gusta o disgusta de Minix, como mi SO se le parece un poco (la misma disposición física del sistema de archivos (por razones prácticas) entre otras cosas.

Hasta ahora he migrado bash (1.08) y gcc (1.4), y las cosas parecen funcionar. Esto implica que voy a obtener algo práctico en pocos meses y me gustaría saber qué características desea la mayoría de la gente. Todas las sugerencias son bienvenidas, pero no prometo que las vaya a implementar :-).

Linus (torvalds@kruuna.helsinki.fi)

PD: Sí, está libre de todo código Minix y tiene un sistema de archivos Multi-threaded. No es portable (usa el cambio de tareas del 386 etc, y probablemente nunca de soporte a nada más que a los discos rígidos para AT, porqué eso es todo lo que tengo:-).

Lo anterior es la primera mención de LINUX en la red. Es el mensaje en el que Linus Torvalds avisó que estaba desarrollando su SO sin darse cuenta el impacto que en pocos años causaría.

FreeBSD y NetCraft

A FreeBSD muchas veces no se lo considera al momento de comparar sistemas operativos y su vigencia e impacto. Sin embargo está muy arraigado dentro de la comunidad de hosteadores y sigue creciendo. Ha estado ganando cerca de un millón de hostnames y más de medio millón de sitios activos desde Julio de 2003.

Top FreeBSD Hosting Providers - June 2004	
Hosting Provider	Active Sites
Yahoo	266,835
NTT/Verio	175,719
SAVVIS Communications	100,377
Datasync	90,324
Pair.com	82,019
iPowerWeb	81,509

Hoy existen cerca de 2.5 millones de sitios activos que corren bajo FreeBSD.

La razón es que FreeBSD es utilizado por operadores de proveedores de hosting compartido, donde miles o aún cientos de miles de sitios se administran colectivamente como parte de un solo sistema.

FreeBSD ha sido sinónimo de hosting compartido en gran escala desde el nacimiento de la web y continúa su relación simbiótica con las compañías más grandes como se puede ver en la figura.

Más de la mitad de los sitios hosteados bajo FreeBSD están en una de las 20 más grandes empresas proveedoras de hosting y mucho del incremento ha sido debido a el hosting ofrecido por Yahoo. Yahoo tiene gran conexión con el proyecto FreeBSD y le ofrece hosting a los servidores ligados al proyecto.



Welcome to The NetBSD Project ``Of course it runs NetBSD."



NetBSD

Alrededor de 1993, While Jolitz y otros colaboradores estaban enfocados en 386BSD. Otros, comenzaron a impacientarse con el lento progreso que presentaba el desarrollo del sistema operativo y tomaron partido por comenzar un desarrollo paralelo tomando lo que originalmente fuera el proyecto 386BSD y el recientemente lanzado Net/2 para portarlo a la plataforma Macintosh. Pronto, derivó en un esfuerzo extra para portarlo también a plataformas Atari ST, Amiga y PC i386.

El desarrollo normal del proyecto encontró muchos seguidores y generó un crecimiento tan grande de NetBSD que prontamente quedó claro qué lugar ocuparía cada uno: FreeBSD quedaría preparado y disponible para plataformas i386 y NetBSD lo haría con las plataformas restantes.

Actualmente la focalización de NetBSD apunta a proveer un sistema multiplataforma y estable orientado a la investigación. La portabilidad de esta versión está asegurada merced a las 33 plataformas en las que puede encontrarse funcionando y su efectividad queda demostrada cuando además de las plataformas disponibles nos encontramos con una versatilidad técnica que lo coloca en igualdad de condiciones entre los modernos equipos con la más alta tecnología y sobre arquitecturas de equipos Intel, Alpha o SPARC debiendo sumar a esto la posibilidad de utilizarlo también en plataformas más antiguas como DEC VAX o Apple Macintosh con procesadores Motorola 68K.

Además, como todo puede utilizarse una y otra vez, sirvió de base para la generación del esquema de portación de FreeBSD a la plataforma Alpha.

Con esto en carrera, el desarrollo de NetBSD no podía hacer más que brindar orientación y conocimiento a otro seguidor de reciente aparición: Linux.

Linux, tal y como lo conocemos hoy, se apoyó en el sistema de arranque de NetBSD para la plataforma Macintosh 68K y desarrolló lo que luego sería el sistema de arranque que permite iniciar una distribución de Linux en dicha plataforma. Pero fundamentalmente el sistema operativo que se vio más beneficiado con la implementación de NetBSD fue sin lugar a dudas OpenBSD para el que puede decirse se trató de un verdadero trampolín.

La historia de FreeBSD y porqué no está tan difundido como Linux.

Uno de los primeros proyectos de sistemas operativos Open Source fue FreeBSD.



Es un descendiente directo del proyecto open source original BSD (Berkeley Software Development) escrito por la Universidad de Berkeley.

En la actualidad existen tres diferentes derivados de BSD: NetBSD, OpenBSD y FreeBSD.

El sistema operativo de Apple OS X (Darwin) está basado en FreeBSD.

El proyecto FreeBSD está realizado por una organización sin fines de lucro y por voluntarios. No tiene un gran presupuesto y por ende no realiza investigaciones sobre cuánto es usado. Pero, sí se puede decir que tiene gran difusión en la industria del web-hosting, especialmente en el segmento de servidores en cluster.

De acuerdo al Open Source Development Labs (OSDL) FreeBSD está en un camino diferente al de Linux. El crecimiento de FreeBSD no es a expensas de Linux. Linux ha heredado un gran número de su código de BSD.

Aunque la comunidad BSD ha estado más tiempo funcionando, es hoy una comunidad más pequeña que Linux. La razón: la fragmentación de la comunidad UNIX y el tipo de licencia.

Aunque la licencia FreeBSD es sencilla de implementar, la licencia GPL (usada por Linux) crea una sensación de comunidad. GPL y las licencias relacionadas han sostenido unida a la comunidad garantizando una continuidad del código.

La actividad de la comunidad alrededor de Linux a finales de 1990 y el soporte de parte de empresas de sistemas y grandes empresas de desarrollo de software ha gestado grandes mejoras en Linux. Por ejemplo mejoras de linux en symmetric multiprocessing (SMP) virtual memory, asynchronous I/O, un "native POSIX thread library", y otras mejoras, junto al soporte de multiples marcas hizo a FreeBSD una elección menos tentadora para grandes cargas de trabajo empresarial.

Sin embargo, FreeBSD permanece siendo un bastión en la infraestructura que mantiene funcionando a Internet. Dice el fundador de ISC (Internet System Consortium) , el grupo que produce BIND (la herramienta dominante de Internet): "Por un lado aplaudo a Linux por aparecer tarde al juego y creando una industria robusta basada en conceptos Open Source".

"Aún más, ISC hostea el servidor que distribuye el Kernel Linux (kernel.org) como un modo de ayudar a la comunidad Linux a continuar su impulso". Por otro lado usamos FreeBSD exclusivamente para el f-root (ver http://www.isc.org/index.pl?/ops/f-root) (ahora en 21 ciudades, usualmente con tres servidores por ciudad) y todos nuestros otros servidores y desarrollo interno"

"Nos gusta la edad de la plataforma. BSD ha existido desde los fines de los 70 y FreeBSD es extremadamente refinado y maduro".

El ISC también hostea el proyecto NetBSD completo, un espejo de OpenBSD y el unico espejo disponible FreeBSD bajo Ipv6. En síntesis FreBSD es una parte crítica de la infraestructura de ISC.

Como resultado de un trabajo acumulativo de años, FreBSD ha liberado recientemente su versión 4.10.

FreBSD 5.x es un desarrollo con nueva tecnología y se espera que sea la nueva versión estable de FreeBSD pronto (llamada FreeBSD 5-STABLE).

Certificaciones Cisco

Hoy es cada vez mayor la necesidad de certificarse en áreas específicas del mundo IT, para acreditar en forma fehaciente los dominios adquiridos a partir de una capacitación adecuada.

Introducción

Podemos conocer mucho sobre tal dispositivo o tal tarea a realizar pero es importante demostrarlo concretamente y de una forma inequívoca. A la hora de postularse para un empleo es cierto que corren con más ventaja aquellas personas que, en igualdad de condiciones, posean certificaciones otorgadas en función de los conocimientos adquiridos. La realidad es que cada vez se presentan más exigencias a la hora de seleccionar personal dada la cantidad creciente de personas que poseen certificaciones. Las empresas exigen más y más conocimientos hasta el punto que el que más sabe, gana. En todo orden siempre ha sido. así. Hace casi diez años, saber utilizar Windows 95 era conveniente para postularse en los primeros lugares de la búsqueda laboral, y se corría con ventaja frente a quienes todavía utilizaban Windows 3.1.

Hoy se plantean alternativas muy similares pero a distinta escala. Certificaciones como MCSE son casi siempre solicitadas por empresas que tienen su infraestructura bajo sistemas operativos de Microsoft. Hoy ya no basta con conocer a fondo un producto, un sistema operativo o una tarea específica. Es también importante, demostrar tener un conocimiento global del tema. Esto en cierta medida garantiza a un empleador que poseemos ciertos conocimientos que él puede verificar, conociendo los contenidos que plantea la certificación.

Tal vez conozcamos un sistema operativo por haber trabajado con él administrando cuentas de usuarios y automatizando las operaciones de backup, pero jamás haber tenido oportunidad de trabajar con él en el área de impresión. Por tal motivo, las certificaciones avalan un conocimiento completo contemplando temas que son comunes para todos aquellos que la hayan obtenido. Podremos saber más de lo que propone la certificación pero no menos para obtenerla. Con esto un empleador se asegura de que, conociendo la certificación, seamos aptos o no para realizar una tarea en particular. Ser certificados demuestra nuestro interés y esfuerzo en permanecer actualizados en tecnologías tan cambiantes. Esto nos reportará beneficios a nosotros y a la empresa que nos desee emplear o donde estemos trabajando.

Certificaciones de CISCO Systems

Desde fines de la década del ´90 la empresa estadounidense Cisco Systems, líder en productos y servicios de networking, desarrolla un plan de estudios para capacitarse en diferentes áreas de la tecnología de networking. Lanzado primero en Estados Unidos, el CNAp, siglas de **C**isco **N**etworking **A**cademy **p**rogram, propone una serie de carreras certificables para adquirir conocimientos específicos en el área de tecnología de redes. La idea siempre es capacitar profesionales que sean aptos para el mantenimiento, diseño e implementación de redes con diferentes niveles de complejidad basadas en tecnologías Cisco. No se trata de capacitación sobre productos exclusivamente ya que los fundamentos deben ser los mismos sea cual fuere la marca utilizada.

Argentina fue el primer país fuera de Estados Unidos en lanzar el CNAp a partir de la selección, por parte de Cisco Systems, de **Fundación Proydesa** como Academia Regional, una ONG que desde 1989 viene implementando tecnología aplicada a la educación. A través de convenios con empresas propietarias de contenidos, Proydesa se encarga del área educacional de las mismas. Desde el año 1998 **Fundación Proydesa** mantiene un acuerdo con Cisco Systems para desarrollar el Cisco Networking Academy program en Argentina. Fundación Proydesa es la Regional Academy de CNAp para la región comprendida por Argentina, Bolivia, Paraguay, Perú y Uruguay. En Argentina existen 41 Academias que, coordinadas por Fundación Proydesa, desarrollan algunas de las carreras que propone el CNAp.

Mantuvimos una entrevista con el Sr. Oscar Gerometta, quién se autodefine como "instructor de instructores" (pues se dedica entre otras cosas a la capacitación de instructores de las diversas carreras), que nos proporcionó información acerca de las carreras que pueden cursarse en Fundación Proydesa y demás academias.

CNAp plantea una serie de carreras certificadas que conducen hacia la certificación máxima otorgada por la empresa: **C**isco **C**ertified **I**nternetwork **E**xpert o simplemente **CCIE**. Como comentaremos más adelante, para certificarse como CCIE no es necesario cursar ninguna carrera. CCIE es la cima de una pirámide de estudio cuyas bases se fundan en conocimientos básicos y avanzados.

Carreras de nivel asociado

Las bases de estudio de esta pirámide plantean conocimientos básicos de networking en general.

Cisco Certified Network Associate (CCNA)

CCNA se plantea como una de las posibles bases de estudio para alcanzar certificaciones de orden superior. Un profesional certificado como CCNA se encuentra en condiciones de instalar, configurar y operar redes de área local (LAN) y de área amplia (WAN) de mediana envergadura.

Figura 1. CCNA



CCNA es la única carrera disponible en todas las academias de la región. Una de las ventajas de esta carrera es que no requiere conocimientos previos, aunque sí es conveniente poseer un manejo básico de computadoras. La carrera completa consta de cuatro módulos teórico-prácticos de un semestre cada uno coordinados por un instructor preparado para tal fin. Con cada módulo completado se recibe un certificado en el cual consta el nivel aprobado. Una de las características de esta modalidad de estudio es que uno puede comenzar un módulo en una academia, finalizarlo y continuar el siguiente en otra, en otro momento y en cualquier lugar del mundo donde se dicte la carrera ya que es de nivel internacional. Posee material de estudio al cual se accede conectado a Internet mediante autenticación con un nombre de usuario y contraseña que posee cada alumno de la academia.

Debemos aclarar que la certificación obtenida al cabo de cada uno de los cuatro semestres es de nivel internacional. En Argentina puede obtenerse una certificación de nivel internacional a través de *testing centers* como **V**irtual **U**niversity **E**nterprises (VUE) o Prometric. Existe un acuerdo entre VUE y Fundación Proydesa por el cual sólo aquellos alumnos que hayan cursado en alguna de las Academias de la red, podrán rendir su examen de certificación en Fundación Proydesa.

Es interesante destacar que, si bien el alumno escoge el *testing center* que prefiera para rendir su examen de certificación, Fundación Proydesa pone a disposición –sólo para aquellos alumnos que hayan cursado en Academias de la red– dos modalidades de *testing centres*, fijo o móvil: Aquellos alumnos residentes en Buenos Aires y alrededores podrán rendir las certificaciones del programa en Fundación Proydesa o tendrán la posibilidad de hacerlo en cualquier otro *testing center* y aquellos que residan en el interior del país tendrán la posibilidad de que se les acerque el *testing center* a la Academia Local de la red donde estudió el alumno o por supuesto en cualquier otro

testing center autorizado . En este punto debemos aclarar que no es necesario haber cursado en una academia para poder presentarse a rendir el examen de certificación CCNA.

El Sr. Gerometta nos decía que *"si bien el examen CCNA no requiere experiencia previa, corre con ventaja el que cursó en una academia para rendir el examen de certificación"*. El examen de certificación CCNA consta de un único examen teórico en el cual coexisten instancias prácticas desarrolladas en software simuladores, descripción de casos y se pide su resolución. El examen incorpora preguntas que para responderlas es recomendable haber ensayado sobre dispositivos directamente, por tal motivo es aconsejable cursar previamente los cuatro módulos en una academia.

Para aquellos alumnos que hayan cursado CCNA en una academia oficial y desean prepararse para el examen de certificación internacional, Proydesa ha desarrollado **Fast Track** (cuya traducción literal sería seguimiento rápido). Fast Track no se define como una carrera sino como un acompañamiento intensivo que ayuda a la preparación del examen de certificación que, a través de la guía de un instructor, se despejan dudas y se aclaran conceptos simulando exámenes similares al de certificación. No se aprenden conceptos nuevos, lo que se busca es adquirir detalles y precisiones necesarias para llevar adelante el examen. *"... en Fast Track no venimos a aprender networking, suponemos que ya sabemos y venimos a refrescar conocimientos, a sacarnos dudas"*, dice Gerometta.

La certificación CCNA internacional debe revalidarse cada tres años.

Cisco Certified Design Associate (CCDA)

Otra de las carreras de nivel asociado que pueden certificarse para llegar a un nivel superior es CCDA. Los profesionales certificados CCDA se encuentran aptos para el diseño de infraestructuras de redes enrutadas y conmutadas. Si bien esta carrera no supone conocimientos previos, para certificarse es necesaria una base de conocimientos como los adquiridos en CCNA. Por tal motivo para ascender en la pirámide y adquirir una certificación de orden superior puede obtenerse tanto la certificación CCNA o CCDA, ya que ambas poseen los mismos fundamentos básicos. Esta carrera no se puede cursar pero sí certificar en Argentina a través de un *testing center* autorizado. La forma de prepararse para obtener la certificación es mediante auto-estudio con bibliografía adecuada, aunque Fundación Proydesa desarrolló una carrera a nivel nacional, **N**etwork **D**esign **P**rogram (NDP), cuyos contenidos se encuentran al nivel de los requeridos para obtener la certificación CCDA.

Figura 2. CCDA



Carreras de nivel profesional

Estas carreras suponen un nivel de conocimiento mayor que el obtenido en el nivel asociado. Estas carreras se ubican como segundo escalón de la pirámide de estudios.

Cisco Certified Network Professional (CCNP)

CCNP es la única carrera de nivel profesional que puede cursarse y certificarse en Argentina, aunque sólo en algunas academias. CCNP consta de cuatro módulos de un semestre cada uno de los cuales se evalúan niveles avanzados de conocimientos en áreas específicas tales como enrutamiento avanzado, conmutación avanzada, procesos de acceso remoto y resolución de fallos de nivel 2 y 3 adquiriéndose un nivel de detalle mayor que en CCNA y desarrollando técnicas aplicables a redes de mayor envergadura. Una de las ventajas de CCNP es que cada módulo puede no ser correlativo, pudiéndose cursar cada uno de ellos en diferente orden debiendo quedar el cuarto obligatoriamente en última instancia. Para acceder a CCNP, es condición indispensable ser CCNA en primer lugar.

La certificación internacional CCNP consta en principio de cuatro exámenes teóricos, uno por cada módulo en cuestión, que pueden tomarse en cualquier orden. Existe una variante de rendir sólo dos exámenes, uno que engloba los primeros tres módulos CCNP y otro que abarca los temas del cuarto módulo. Todos los exámenes son certificables en Argentina a través de los *testing centers* citados anteriormente.

Figura 3. CCNP



Cisco Certified Design Professional (CCDP)

CCDP supone un conocimiento avanzado en el área de diseño de redes de área local y área amplia aplicando prácticas de diseño modular asegurando soluciones óptimas para las necesidades técnicas y comerciales. A pesar de que para obtener la certificación CCDP, un profesional deber ser previamente certificado CCNA o CCDA, esta carrera no se puede cursar pero sí certificar en Argentina a través de un

testing center autorizado. La forma de prepararse para obtener la certificación es mediante auto-estudio con bibliografía adecuada. Los exámenes de certificación son tres, pudiéndose rendir uno que abarque dos módulos y otro, el restante.

Figura 4. CCDP

Cisco Certified Security Professional (CCSP)

CCSP indica un nivel avanzado en seguridad de redes Cisco. Un profesional certificado como CCSP se encuentra apto para ase-



gurar y administrar infraestructuras de redes para mejorar la productividad y reducir costos. El contenido de CCSP incluye seguridad perimetral, redes privadas virtuales, protección de intrusiones y la combinación de estas técnicas para lograr una solución de seguridad integrada. La certificación CCSP requiere ser certificado CCNA o CCIP, que veremos a continuación. Los exámenes de certificación son cinco, uno por cada módulo, no pudiéndose rendir agrupados. Otra vez, a pesar de no poder cursarse una carrera que desarrolle los temas requeridos para preparar el examen de certificación, en Argentina sí se puede certificar a través de los *testing centers* autorizados para tal fin, debiéndose preparar el examen con bibliografía adecuada.

Figura 5. CCSP

Cisco Certified Internetwork Professional (CCIP)

Figura 6. CCIP

CCIP es una certificación que provee habilidades para el trabajo en organizaciones



proveedoras de servicios (ISP) con competencias en la solución de networking de infraestructura de redes IP, incluyendo calidad de servicio, redes conmutadas multicapa, enrutamiento IP avanzado, etc. Es una certificación que requiere



ser certificado CCNA previamente. Los exámenes de certificación son cuatro, pudiéndose rendir uno que agrupe dos módulos y los otros dos exámenes por separado. Respecto a la certificación en Argentina, es lo mismo que para los otros exámenes de certificación.

Carreras de nivel experto

Cisco Certified Internetwork Expert (CCIE)

Figura 7. CCIE

Y llegamos a la cúspide de la pirámide de estudios, hacia la cual conducen todos los caminos tomados.

CCIE es la certificación de alto nivel más respetada, reconocida mundialmente como un "doctorado" de networking. Según Cisco, menos del 3% de la cantidad de profesionales



certificados llega a ser CCIE. No podemos saber con certeza cuántos CCIEs hay en Argentina ya que muchos de ellos seguramente trabajan en el exterior, aunque sería más correcto preguntarse cuántos CCIEs argentinos hay. Una cifra tentativa podría indicar que la cantidad de profesionales CCIE argentinos no supera los 15.

Anteriormente dijimos que no era necesario cursar ninguna carrera para certificarse como CCIE, esto es así pues es considerada como una certificación de auto-estudio. Certificarse como CCIE no involucra haber cursado ninguna cantidad de tiempo ni haber obtenido una certificación de orden asociado o profesional.

Uno puede pagar los exámenes y rindiéndolos correctamente, obtener la certificación. Pero esto no es tan así, pues dichos exámenes involucran conocimientos de tan alto nivel que los postulantes generalmente poseen certificaciones de niveles inferiores al CCIE. Si bien no es imprescindible, es altamente recomendable haber cursado carreras anteriores. Según Cisco, la mejor preparación es la experiencia, recomendando de 3 a 5 años mínimos antes de presentarse.

No existe una única rama de especialización CCIE ; según la trayectoria segui-



da, existen especializaciones en el área de conmutación y enrutamiento, proveedor de servicios, seguridad y tecnologías de voz sobre IP.

La instancia de certificación CCIE abarca dos exámenes : uno teórico y uno absolutamente práctico.

El examen teórico puede ser certificado en Argentina a través de *testing centers* autorizados como VUE o Prometric. Su valor ronda los US$ 300 y consta de unas 100 preguntas aproximadamente que se deberán contestar en un tiempo inferior a los 120 minutos. En caso de resultar desaprobado, podrá darse nuevamente esperando al menos 72 horas entre intentos. Dentro de los 18 meses de haberse aprobado el examen teórico deberá intentarse dar el examen de carácter práctico.

El examen práctico consta de dos etapas : una de configuración de dispositivos según un laboratorio armado para tal fin y otra etapa donde se deben detectar fallas causadas intencionalmente y corregirlas. Todo esto supone una permanencia de 8 horas de laboratorio. El único prerrequisito es haber aprobado la instancia teórica para poder presentarse al examen práctico.

El examen práctico no se puede rendir en Argentina. Según la especialización elegida, existen sedes en Brasil y Estados Unidos. Su costo ronda los US$ 1250 sin incluir gastos de pasaje y estadía, cuyo precio se deberá ajustar según el concepto de impuestos locales. La certificación CCIE es la única que requiere un examen de carácter práctico para lograrla.

En Argentina, Cisco montó un laboratorio para aquellos que se postulen a CCIE y deseen entrenarse. La condición para asistir es haber aprobado el examen teórico.

Cisco Qualified Specialist

Son módulos autónomos que permiten desarrollar aptitudes en áreas específicas. No llegan a ser carreras dada su corta duración. Entre las certificaciones se encuentran: telefonía IP, tecnologías ópticas, medios de almacenamiento de red, redes inalámbricas, seguridad y VPNs, etc.



Figura 8. Cisco Qualified Specialist

En Fundación Proydesa y demás Academias Locales se desarrollan otras carreras de uno o dos módulos semestrales surgidas por convenios con otras empresas propietarias de contenidos. Estas empresas auspician el desarrollo de los respectivos contenidos. Entre ellas se encuentran:

>>Fundamentals of Unix y **Fundamentals of Java Programming**.

>>IT Essentials I Hardware & Software y **IT Essentials II Network Operative Systems** cuyos contenidos son sponsoreados por Hewlett-Packard.

>>Linux Systems Administration cuyos contenidos son sponsoreados por Red Hat.

>>Fundamentals of Voice and Data Cabling sponsoreada por Panduit.

>>Administración de base de datos Oracle sponsoreada por Oracle University.

>>Network Design Program y **Enterprise Security & Risk** son dos carreras desarrolladas íntegramente por Fundación Proydesa sin sponsor de empresas que, a través del Consejo Profesional de Ingeniería de Telecomunicaciones, Electrónica y Computación (COPITEC), puede obtenerse una certificación de carácter nacional. Ambas poseen los contenidos apropiados para presentarse a rendir las certificaciones internacionales **CCDA** y **CompTIA Security +** respectivamente.

>>Fundamentals of Wireless LAN, de reciente aparición, esta carrera brinda los conocimientos requeridos para obtener la certificación Wireless Lan Specialist, módulo autónomo Cisco Qualified Specialist.

Nota: Las ilustraciones son propiedad exclusiva de Cisco Systems

Para mayor información visite www.cisco.com, www.pearsonvue.com, www.prometric.com

Fundación Proydesa Sarmiento 930 3º piso Capital Federal, teléfonos 4326-4917 / 5958 / 9839
Website: http://www.proydesa.org e-mail: academia@proydesa.org

Conclusiones

Tenemos certificaciones para todos los gustos. La realidad en Argentina demuestra que muchas veces no se valora tener profesionales certificados. Las empresas deben evaluar hasta qué punto les conviene mantener técnicos altamente especializados y certificados. Muchas empresas multinacionales que residen en el país traen su mano de obra e infraestructura ya configurada del exterior, con lo cual podemos sentir cierto recelo al respecto. No es muy viable hablar de sueldos de acuerdo a la certificación obtenida pero lo que sí estamos en condiciones de afirmar es que quien se certifica está en condiciones de adquirir un plus salarial.

Deseamos expresar nuestro más especial agradecimiento al Sr. Oscar Gerometta, Instructor de **Fundación Proydesa**, por su valiosa colaboración y predisposición para la realización de esta nota.

Leonel F. Becchio & Ximena Antona



Microsoft



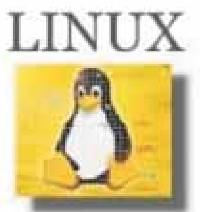
Security



WEB Design



LINUX



Suplemento Guía Cursos y Carreras - 1 Agosto 2004 a 31 Julio 2005



www.cortech.com.ar

Garantía de Educación

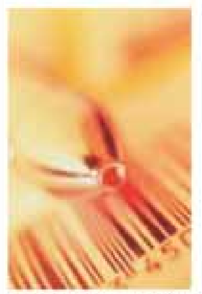


Un alumno, cuando "compra" un curso no busca otra cosa más que ADQUIRIR UN APRENDIZAJE, capacitarse, aprender, y crecer en el mundo de IT conociendo siempre las últimas tecnologías. La Garantía de aprendizaje COR, permite a TODOS los alumnos re-cursar cuantas veces sea necesario las Carreras o Cursos de COR TECH.

Aprovechá la posibilidad de volver a hacer tus cursos; ya sea si te quedaste con dudas, o faltaste alguna que otra clase, o simplemente porque deseas conocer el perfil de otro profesor o volver hacer el curso para conocer gente nueva.

En COR no comprás UN CURSO; comprás UN APRENDIZAJE (y queremos asegurarnos de dártelo).

Garantía de Precio



COR Technologies garantiza ofrecerte un precio 10 % más bajo para cualquier presupuesto de Educación o consultoría en Buenos Aires o en el Interior del País.

Presentando el presupuesto de la competencia (ya sea por escrito o por e-mail; para un mínimo de 2 Alumnos ó mínimo de $1000) COR te brinda la Capacitación o la Solución buscada a un costo 10 % mas bajo (sólo multiplicá tu presupuesto por 0,90 y obtené el precio que te ofrece COR).

La Garantía de precio será respetada siempre y cuando el precio cobrado finalmente no sea menor al costo de realizar la Capacitación / Consultoría. COR Technologies se reserva el derecho de validación de los presupuestos propiamente presentados.

Garantía de Consultoría



Una característica de nuestras consultorías es que COR garantiza la completa conformidad de Cliente; o se reintegra el monto total de lo abonado para la misma. La Garantía de Consultoría permite al cliente estar confiado de que recibirá lo que desea; y que COR garantiza el resultado del problema mediante la Solución oportunamente propuesta.

COR Cheks



COR premia la Capacitación entregándole a todos nuestros alumnos la suma correspondiente de CORCheks. Cada CORChek es equivalente a un Peso; para ser utilizado en cualquiera de nuestros Cursos y Carreras. Los CORCheks no son transferibles; y no pueden utilizarse junto a otras promociones.

Microsoft Certified Systems Administrator (MCSA) Windows 2003

Microsoft CERTIFIED Technical Education Center

Microsoft CERTIFIED Partner for Learning Solutions

EXAMEN - Client	CURSO - Client
Examen 70–270: Installing, Configuring, and Administering Microsoft Windows XP Professional	**Curso 2285:** Installing, Configuring, and Administering Microsoft Windows XP Professional (Duración **16 hs**)
EXAMEN - Networking	**CURSO - Networking**
Examen 70–290: Managing and Maintaining a Microsoft Windows Server 2003 Environment	**Curso 2273:** Managing and Maintaining a Microsoft Windows Server 2003 Environment (Duración **40 hs**)
Examen 70–291: Implementing, Managing, and Maintaining a Microsoft Windows Server 2003 Network Infrastructure	**Curso 2276:** Implementing a Microsoft Windows Server 2003 Network Infrastructure: Network Hosts (Duración **16 hs**)
	Curso 2277: Implementing, Managing, and Maintaining a MS Windows Server 2003 Network Infrastructure: Network Services (Duración **40 hs**)
EXAMEN - Elective	**CURSO - Elective**
Examen 70-227: Installing, Configuring, and Administering Microsoft Internet Security and Acceleration (ISA) Server 2000, Enterprise Edition	**Curso 2159:** Deploying and Managing Microsoft Internet Security and Acceleration Server 2000 (Duración **24 hs**)

Cursos: 5 (cinco) | **MOC's incluidos: 5 (cinco)** | **Duración Total: 136 hs**



Microsoft Certified Systems Administrator (MCSA Sec.) Security on Windows 2003 // Track Recomendado //

EXAMEN - Client	CURSO - Client
Examen 70–270: Installing, Configuring, and Administering Microsoft Windows XP Professional	**Curso 2285:** Installing, Configuring, and Administering Microsoft Windows XP Professional (Duración **16 hs**)
EXAMEN - Networking	**CURSO - Networking**
Examen 70–290: Managing and Maintaining a Microsoft Windows Server 2003 Environment	**Curso 2273:** Managing and Maintaining a Microsoft Windows Server 2003 Environment (Duración **40 hs**)
Examen 70–291: Implementing, Managing, and Maintaining a Microsoft Windows Server 2003 Network Infrastructure	**Curso 2276:** Implementing a Microsoft Windows Server 2003 Network Infrastructure: Network Hosts (Duración **16 hs**)
	Curso 2277: Implementing, Managing, and Maintaining a MS Windows Server 2003 Network Infrastructure: Network Services (Duración **40 hs**)
EXAMEN - Elective	**CURSO - Elective**
Examen 70-227: Installing, Configuring, and Administering Microsoft Internet Security and Acceleration (ISA) Server 2000, Enterprise Edition	**Curso 2159:** Deploying and Managing Microsoft Internet Security and Acceleration Server 2000 (Duración **24 hs**)
Examen 70-299: Implementing and Administering Security in a Microsoft Windows Server 2003 Network	**Curso 2823:** Implementing and Administering Security in a Microsoft Windows Server 2003 Network (Duración **40 hs**)

Cursos: 6 (seis) | **MOC's incluidos: 6 (seis)** | **Duración Total: 176 hs**

Fechas Inicio Calendario

MCSA y MCSA Security

INICIO	DIAS	HORARIO
03-08-04	M-J	9.00 a 13.00
13-08-04	L-M-V	18.30 a 22.30
12-08-04	M-J	18.30 a 22.30
03-09-04	L-M-V	9.00 a 13.00
09-09-04	M-J	9.00 a 13.00
08-09-04	L-M-V	18.30 a 22.30
05-10-04	M-J	18.30 a 22.30
12-10-04	M-J	9.00 a 13.00
13-10-04	L-M-V	18.30 a 22.30
02-11-04	M-J	9.00 a 13.00
10-11-04	L-M-V	18.30 a 22.30
15-11-04	L-M-V	9.00 a 13.00
04-01-05	M-J	18.30 a 22.30
14-01-05	L-M-V	18.30 a 22.30
19-01-05	L-M-V	9.00 a 13.00
08-02-05	M-J	9.00 a 13.00
11-02-05	L-M-V	9.00 a 13.00
18-02-05	L-M-V	18.30 a 22.30
15-03-05	M-J	9.00 a 13.00
18-03-05	L-M-V	18.30 a 22.30
10-03-05	M-J	18.30 a 22.30
19-04-05	M-J	18.30 a 22.30
14-04-05	M-J	9.00 a 13.00
13-04-05	L-M-V	9.00 a 13.00
04-05-05	M-J	18.30 a 22.30
10-05-05	M-J	9.00 a 13.00
18-05-05	L-M-V	18.30 a 22.30
07-06-05	M-J	9.00 a 13.00
15-06-05	L-M-V	18.30 a 22.30
17-06-05	L-M-V	9.00 a 13.00
05-07-05	M-J	18.30 a 22.30
13-07-05	L-M-V	18.30 a 22.30
15-07-05	L-M-V	9.00 a 13.00

Para más información sobre la carrera MCSA Windows 2003 visitá www.cortech.com.ar/ms/mcsa.htm ó www.microsoft.com/learning/mcp/mcsa/default.asp

Certificaciones Internacionales

¿Dónde se pueden rendir los exámenes para certificarme como MCSA y/o MCSE?

Podés hacer los exámenes en cualquier centro CTEC (Certified Training Education Center) de tu localidad que provea exámenes VUE: Virtual Universities Enterprise (ver www.vue.com)

Deberás entonces reservar tu turno de examen y abonar el costo correspondiente de 125.00 U$S en U.S.A. por examen; y 80.00 U$S en Argentina (tarifas adicionales o descuentos pueden aplicarse en otras regiones).

Todos los Tracks MCSA

¿Cuáles son los Exámenes que debo tomar para recibirme de MCSA?

Existen muchísimas combinaciones de Exámenes para recibirse de MCSA: Microsoft Certified Systems Administrator. Cada una con diferentes especializaciones y electivos para tomar.

MCSA

http://www.cortech.com.ar/gen/mcsawin2003.pdf
http://www.cortech.com.ar/gen/MCSASec2000-2003.pdf
http://www.cortech.com.ar/gen/MCSAMes2000-2003.pdf

Microsoft Certified Systems Engineer (MCSE) Windows 2003

Microsoft CERTIFIED Technical Education Center — Microsoft CERTIFIED Partner for Learning Solutions

EXAMEN - Client	CURSO - Client
Examen 70–270: Installing, Configuring, and Administering Microsoft Windows XP Professional	**Curso 2285:** Installing, Configuring, and Administering Microsoft Windows XP Professional (Duración **16 hs**)
EXAMEN - Networking	**CURSO - Networking**
Examen 70–290: Managing and Maintaining a Microsoft Windows Server 2003 Environment	**Curso 2273:** Managing and Maintaining a Microsoft Windows Server 2003 Environment (Duración **40 hs**)
Examen 70–291: Implementing, Managing, and Maintaining a Microsoft Windows Server 2003 Network Infrastructure	**Curso 2276:** Implementing a Microsoft Windows Server 2003 Network Infrastructure: Network Hosts (Duración **16 hs**)
	Curso 2277: Implementing, Managing, and Maintaining a MS Windows Server 2003 Network Infrastructure: Network Services (Duración **40 hs**)
Examen 70-293: Planning and Maintaining a Microsoft Windows Server 2003 Network Infrastructure	**Curso 2278:** Planning and Maintaining a Microsoft Windows Server 2003 Network Infrastructure (Duración **40 hs**)
Examen 70-294: Planning, Implementing, and Maintaining a Microsoft Windows Server 2003 Active Directory Infrastructure	**Curso 2279:** Planning, Implementing, and Maintaining a Microsoft Windows Server 2003 Active Directory Infrastructure (Duración **40 hs**)
EXAMEN - Design	**CURSO - Design**
Examen 70-298: Designing Security for a Microsoft Windows Server 2003 Network	**Curso 2830:** Designing Security for Microsoft Networks (Duración **24 hs**)
EXAMEN - Elective	**CURSO - Elective**
Examen 70-227: Installing, Configuring, and Administering Microsoft Internet Security and Acceleration (ISA) Server 2000, Enterprise Edition	**Curso 2159:** Deploying and Managing Microsoft Internet Security and Acceleration Server 2000 (Duración **24 hs**)
# Cursos: 8 (ocho) **MOC's incluidos: 8 (ocho)**	**Duración Total: 240 hs**

Microsoft Certified Systems Engineer (MCSE Sec.) Security on Windows 2003 (Carrera MCSE + Examen 70-299)

Examen 70-299: Implementing and Administering Security in a Microsoft Windows Server 2003 Network	**Curso 2823:** Implementing and Administering Security in a Microsoft Windows Server 2003 Network (Duración **40 hs**)
# Cursos: 9 (nueve) **MOC's incluidos: 9 (nueve)**	**Duración Total: 280 hs**

Microsoft Certified Systems Engineer // Track Recomendado // Security + 2282 on Win. 2003 (Carrera MCSE Security + Examen 70-297)

Examen 70-297: Designing a Microsoft Windows Server 2003 Active Directory and Network Infrastructure	**Curso 2282:** Designing a Microsoft Windows Server 2003 Active Directory and Network Infrastructure (Duración **40 hs**)
# Cursos: 10 (diez) **MOC's incluidos: 10 (diez)**	**Duración Total: 320 hs**



Fechas Inicio Calendario

MCSE, MCSE Security y MCSE Sec + 2282

INICIO	DIAS	HORARIO
10-08-04	M-J	9.00 a 13.00
20-08-04	L-M-V	18.30 a 22.30
19-08-04	M-J	18.30 a 22.30
10-09-04	L-M-V	9.00 a 13.00
16-09-04	M-J	9.00 a 13.00
15-09-04	L-M-V	18.30 a 22.30
12-10-04	M-J	18.30 a 22.30
19-10-04	M-J	9.00 a 13.00
20-10-04	L-M-V	18.30 a 22.30
09-11-04	M-J	9.00 a 13.00
17-11-04	L-M-V	18.30 a 22.30
24-11-04	L-M-V	9.00 a 13.00
11-01-05	M-J	18.30 a 22.30
21-01-05	L-M-V	18.30 a 22.30
26-01-05	L-M-V	9.00 a 13.00
15-02-05	M-J	9.00 a 13.00
18-02-05	L-M-V	9.00 a 13.00
25-02-05	L-M-V	18.30 a 22.30
22-03-05	M-J	9.00 a 13.00
25-03-05	L-M-V	18.30 a 22.30
17-03-05	M-J	18.30 a 22.30
26-04-05	M-J	18.30 a 22.30
21-04-05	M-J	9.00 a 13.00
20-04-05	L-M-V	9.00 a 13.00
10-05-05	M-J	18.30 a 22.30
17-05-05	M-J	9.00 a 13.00
25-05-05	L-M-V	18.30 a 22.30
14-06-05	M-J	9.00 a 13.00
22-06-05	L-M-V	18.30 a 22.30
24-06-05	L-M-V	9.00 a 13.00
12-07-05	M-J	18.30 a 22.30
20-07-05	L-M-V	18.30 a 22.30
22-07-05	L-M-V	9.00 a 13.00

Para más información sobre la carrera MCSE Windows 2003 visitá www.cortech.com.ar/ms/mcse.htm ó www.microsoft.com/learning/mcp/mcse/default.asp

Todos los Tracks MCSE

¿Cuáles son los Exámenes que debo tomar para recibirme de MCSE?

Existen muchísimas combinaciones de Exámenes para recibirse de MCSE: Microsoft Certified Systems Engineer. Cada una con diferentes especializaciones y electivos para tomar.

MCSE

http://www.cortech.com.ar/gen/mcsewin2003.pdf
http://www.cortech.com.ar/gen/MCSESec2000-2003.pdf
http://www.cortech.com.ar/gen/MCSEMes2000-2003.pdf

Logos MCP

¿Cuáles son los logos que podré utilizar cuándo me reciba de MCP, MCSA, MCSE, MCDBA, MCAD ó MCSD?¿Existe alguna diferencia entre los logos con especializaciones en Security, Messaging, etc..?

Al finalizar de haber rendido todos los Exámenes de cada Carrera Microsoft, podrás utilizar el logo correspondiente. Todos las Carreras (como así tambien las especializaciones) poseen un logo diferente.

Podés encontrar todos los logos Microsoft correspondientes en http://www.microsoft.com/learning/mcpexams/faq/logo.asp

Microsoft Certified Application Developer (MCAD) Visual Basic .NET

Microsoft CERTIFIED Technical Education Center

Microsoft CERTIFIED Partner for Learning Solutions

EXAMEN - Módulo I	CURSO - Módulo I
Examen 70-305: Developing and Implementing Web Applications with Microsoft® Visual Basic® .NET and Microsoft® Visual Studio® .NET	**Curso 2559:** Introduction to Visual Basic .NET Programming with Microsoft .NET (Duración **20 hs**)
	Curso 2310: Developing Microsoft ASP.NET Web Applications Using Visual Studio .NET (Duración **40 hs**)
EXAMEN - Módulo II	**CURSO - Módulo II**
Examen 70-310: Developing XML Web Services and Server Components with Microsoft® Visual Basic® .NET and the Microsoft® .NET Framework	**Curso 2415:** Programming with the Microsoft® .NET Framework (Microsoft V. Basic® .NET) (Duración **40 hs**)
	Curso 2524: Developing XML Web Services Using Microsoft® ASP.NET (Duración **20 hs**)
	Curso 2557: Building COM+ Applications Using Microsoft® .NET Enterprise Services (Duración **20 hs**)
EXAMEN - Módulo III	**CURSO - Módulo III**
Examen 70-229: Designing and Impl. Databases with MS SQL Server 2000™ Enterprise Edition	**Curso 2073:** Programming a Microsoft SQL Server 2000 Database (Duración **40 hs**)
# Cursos: 6 (seis) **MOC's incluidos: 6 (seis)**	**Duración Total: 180 hs**

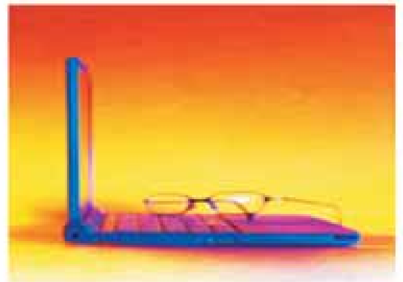
Microsoft Certified Solution Developer (MCSD) Visual Basic .NET (Carrera MCAD + Examen 70-300 + Examen 70-306)

EXAMEN - Módulo IV	CURSO - Módulo IV
Examen 70-300: Analyzing Requirements & Defining .NET Solution Architectures	**Curso 2710 :** Analyzing Requirements and Defining .NET Solution Architecture (Duración **40 hs**)
EXAMEN - Módulo V	**CURSO - Módulo V**
Examen 70-306: Developing & Impementing Windows-based Applications with Microsoft Visual Basic .NET & MS Visual Studio .NET	**Curso 2565:** Developing Microsoft .NET Applications for Windows (Visual Basic .NET) (Duración **20 hs**)
# Cursos: 8 (ocho) **MOC's incluidos: 8 (ocho)**	**Duración Total: 240 hs**

Microsoft Certified Application Developer (MCAD) C#™ .NET

// Track Recomendado //

EXAMEN - Módulo I	CURSO - Módulo I
Examen 70-315: Developing and Implementing Web Applications with Microsoft Visual C#™ .NET and Microsoft Visual Studio .NET	**Curso 2609:** Introduction to C# Programming with Microsoft .NET (Duración **20 hs**)
	Curso 2310: Developing Microsoft ASP.NET Web Applications Using Visual Studio .NET (Duración **40 hs**)
EXAMEN - Módulo II	**CURSO - Módulo II**
Examen 70-320: Developing XML Web Services and Server Components with Microsoft Visual C# and the Microsoft .NET Framework	**Curso 2349:** Programming with the Microsoft .NET Framework (Microsoft Visual C# .NET).(Duración **40 hs**)
	Curso 2524: Developing XML Web Services Using Microsoft® ASP.NET (Duración **20 hs**)
	Curso 2557: Building COM+ Applications Using Microsoft® .NET Enterprise Services (Duración **20 hs**)
EXAMEN - Módulo III	**CURSO - Módulo III**
Examen 70-229: Designing and Impl. Databases with MS SQL Server 2000™ Enterprise Edition	**Curso 2073:** Programming a Microsoft SQL Server 2000 Database (Duración **40 hs**)
# Cursos: 6 (seis) **MOC's incluidos: 6 (seis)**	**Duración Total: 180 hs**



SQL Server

Las dos Certificaciones de SQL más importantes son: **Examen 70-228** (Installing, Configuring, and Administering Microsoft SQL Server 2000 Enterprise Edition) y **Examen 70-229** (Designing and Implementing Databases with Microsoft SQL Server 2000 Enterprise Edition).

Estos Exámenes podrán prepararse con los Cursos Oficiales **2072** (Administering a MS-SQL Server 2000 Database) y **2073** (Programming a MS-SQL Server 2000 Database) respectivamente.

Fechas Inicio Calendario

MCAD V. Basic, MCSD y MCAD C#

INICIO	DIAS	HORARIO
03-08-04	M-J	9.00 a 13.00
13-08-04	L-M-V	18.30 a 22.30
12-08-04	M-J	18.30 a 22.30
03-09-04	L-M-V	9.00 a 13.00
09-09-04	M-J	9.00 a 13.00
08-09-04	L-M-V	18.30 a 22.30
05-10-04	M-J	18.30 a 22.30
12-10-04	M-J	9.00 a 13.00
13-10-04	L-M-V	18.30 a 22.30
02-11-04	M-J	9.00 a 13.00
10-11-04	L-M-V	18.30 a 22.30
15-11-04	L-M-V	9.00 a 13.00
04-01-05	M-J	18.30 a 22.30
14-01-05	L-M-V	18.30 a 22.30
19-01-05	L-M-V	9.00 a 13.00
08-02-05	M-J	9.00 a 13.00
11-02-05	L-M-V	9.00 a 13.00
18-02-05	L-M-V	18.30 a 22.30
15-03-05	M-J	9.00 a 13.00
18-03-05	L-M-V	18.30 a 22.30
10-03-05	M-J	18.30 a 22.30
19-04-05	M-J	18.30 a 22.30
14-04-05	M-J	9.00 a 13.00
13-04-05	L-M-V	9.00 a 13.00
04-05-05	M-J	18.30 a 22.30
10-05-05	M-J	9.00 a 13.00
18-05-05	L-M-V	18.30 a 22.30
07-06-05	M-J	9.00 a 13.00
15-06-05	L-M-V	18.30 a 22.30
17-06-05	L-M-V	9.00 a 13.00
05-07-05	M-J	18.30 a 22.30
13-07-05	L-M-V	18.30 a 22.30
15-07-05	L-M-V	9.00 a 13.00

Para más información sobre la carrera MCAD .NET, MCSD y MCAD C# visitá www.cortech.com.ar/ms/ms4.htm ó www.microsoft.com/learning/mcp/mcad/

Links Microsoft

¿Existe algún link en donde se puedan ver todos los Exámenes actuales de Microsoft y todos sus Cursos Oficiales asociados?

Para ver todos los Exámenes Microsoft vigentes que existen visitá **www.microsoft.com/learning/mcpexams/prepare/findexam.asp** Allí los podrás visualizar por Carreras o por número de Examen.

Y para ver todos los Cursos Oficiales vigentes visitá **www.microsoft.com/traincert/training/find/findcourse.asp** Allí podrás visualizarlos por Producto o por número de Curso.

Carrera MCDBA

¿Cuáles son los exámenes que debo tomar para realizar la Carrera MCDBA?

Para ver el listado completo de todas las opciones que existen para convertirte en Microsoft Certified Data Base Administrator (MCDBA) te recomendamos visitar la siguiente página WEB: http://www.microsoft.com/learning/mcp/mcdba/default.asp.

El Track recomendado para convertirte en MCDBA es realizar la Carrera MCSE de 240 hs de Duración (7 Exámenes) + los Exámenes de SQL Server 70-228 (Administering) y 70-229 (Programming)

Fechas Inicio Calendario
WEB Design Completa y Expert

INICIO	DIAS	HORARIO
06-08-04	L-M-V	9.30 a 12.30
12-08-04	M-J	18.30 a 21.30
17-08-04	M-J	14.00 a 17.00
04-09-04	S	10.00 a 13.00
08-09-04	L-M-V	18.30 a 21.30
16-09-04	M-J	9.30 a 12.30
01-10-04	L-M-V	9.30 a 12.30
07-10-04	M-J	18.30 a 21.30
13-10-04	L-M-V	14.00 a 17.00
06-11-04	S	10.00 a 13.00
10-11-04	L-M-V	18.30 a 21.30
18-11-04	M-J	9.30 a 12.30
07-01-05	L-M-V	9.30 a 12.30
13-01-05	M-J	18.30 a 21.30
18-01-05	M-J	14.00 a 17.00
05-02-05	S	10.00 a 13.00
11-02-05	L-M-V	18.30 a 21.30
17-02-05	M-J	9.30 a 12.30
04-03-05	L-M-V	9.30 a 12.30
17-03-05	M-J	18.30 a 21.30
18-03-05	L-M-V	14.00 a 17.00
09-04-04	S	10.00 a 13.00
08-04-05	L-M-V	18.30 a 21.30
21-04-05	M-J	9.30 a 12.30
13-05-04	L-M-V	9.30 a 12.30
12-05-05	M-J	18.30 a 21.30
19-05-05	M-J	14.00 a 17.00
11-06-05	S	10.00 a 13.00
15-06-05	L-M-V	18.30 a 21.30
16-06-05	M-J	9.30 a 12.30
06-07-05	L-M-V	9.30 a 12.30
14-07-05	M-J	18.30 a 21.30
20-07-05	L-M-V	14.00 a 17.00

Carrera WEB Design Completa
WEB1 + WEB2 + WEB3

EXAMEN - WEB Design	CURSO - WEB Design
Examen Dreamweaver MX 2004 Designer	**Módulo WEB1:** Curso de Front Page XP y Macromedia Dreamweaver MX 04 (Duración **18 hs**)
Exámenes Flash MX 2004 Designer y Developer	**Módulo WEB2:** Curso de Macromedia Flash MX 04 y Macromedia Fireworks MX 04 (Duración **21 hs**)
Exámenes Dreamweaver MX 2004 Designer y Developer	**Módulo WEB3:** Curso de Edición HTML e Introd. a Programación ASP (Duración **21 hs**)

Cursos: 3 (tres) **WOG's incluidos: 1 (uno)** **Duración Total: 60 hs**

Carrera WEB Design Expert
WEB1 + WEB2 + WEB3 + WEB4 + WEB5 // Track Recomendado //

EXAMEN - WEB Design	CURSO - WEB Developer
Examen Dreamweaver MX 2004 Developer	**Módulo WEB4:** Curso Programación ASP Avanzado (Duración **21 hs**)
-- --	**Módulo WEB5:** Curso Programación PHP Avanzado (Duración **21 hs**)

Cursos: 5 (cinco) **WOG's incluidos: 2 (dos)** **Duración Total: 102 hs**

Para más información sobre la carrera WEB Design Completa y WEB Design Expert visitá www.cortech.com.ar/web/web1.htm



Fechas Inicio Calendario
Carrera COR Security + Especializaciones

INICIO	DIAS	HORARIO
17-08-04	M-J	9.00 a 13.00
27-08-04	L-M-V	18.30 a 22.30
26-08-04	M-J	18.30 a 22.30
17-09-04	L-M-V	9.00 a 13.00
23-09-04	M-J	9.00 a 13.00
22-09-04	L-M-V	18.30 a 22.30
19-10-04	M-J	18.30 a 22.30
26-10-04	M-J	9.00 a 13.00
27-10-04	L-M-V	18.30 a 22.30
16-11-04	M-J	9.00 a 13.00
24-11-04	L-M-V	18.30 a 22.30
29-11-04	L-M-V	9.00 a 13.00
18-01-05	M-J	18.30 a 22.30
28-01-05	L-M-V	18.30 a 22.30
02-02-05	L-M-V	9.00 a 13.00
22-02-05	M-J	9.00 a 13.00
25-02-05	L-M-V	9.00 a 13.00
04-03-05	L-M-V	18.30 a 22.30
29-03-05	M-J	9.00 a 13.00
23-03-05	L-M-V	18.30 a 22.30
24-03-05	M-J	18.30 a 22.30
14-04-05	M-J	18.30 a 22.30
28-04-05	M-J	9.00 a 13.00
27-04-05	L-M-V	9.00 a 13.00
17-05-05	M-J	18.30 a 22.30
24-05-05	M-J	9.00 a 13.00
01-06-05	L-M-V	18.30 a 22.30
21-06-05	M-J	9.00 a 13.00
29-06-05	L-M-V	18.30 a 22.30
01-07-05	L-M-V	9.00 a 13.00
19-07-05	M-J	18.30 a 22.30
27-07-05	L-M-V	18.30 a 22.30
29-07-05	L-M-V	9.00 a 13.00

Carrera COR Security // Track Recomendado //
SEC1 + SEC2 + Especialización (a elección)

EXAMEN - CISSP	CURSO - Security
 CISSP: Certified Information Systems Security Professional	**Clínica SEC1:** Seguridad y sus fundamentos (Duración **20 hs**) **Clínica SEC2:** Seguridad Avanzada (Duración **20 hs**)
Especialización LINUX	**Especialización Microsoft**
Módulo LX5: Seguridad y contra-seguridad en Redes (Duración **12hs**) + **Workshop LX6:** Workshops Servidor de Firewall y Squid (Comparación con ISA Server) (Duración **12 hs**) + **Workshop LX8:** Workshops Implementando VPNs bajo Linux (Duración **12 hs**)	**Curso 2159:** Deploying and Managing Microsoft Internet Security and Acceleration Server 2000 (Duración **24 hs**) + **Curso 40 hs Seguridad Electivo** de la Currícula Oficial Microsoft + **Curso 2823:** Implementing and Administering Security in a Microsoft Windows Server 2003 Network (Duración **40 hs**)
Incluye Material **# Cursos: 5 (cinco)** **Duración Total: 76 hs**	**Incluye Material** **# Cursos: 5 (cinco)** **Duración Total: 144 hs**

Para más información sobre la carrera COR Security y sus Especializaciones visitá www.secure105.com.ar

Cursos Intensivos y Personalizados

¿Cómo puedo hacer para que yo o la gente de mi Empresa pueda cursar cualquiera de los Cursos y Carreras Microsoft, Security, WEB Design o Linux de manera Personalizada / Intensiva?

Te recomendamos averiguar por costos y metodologías de cursada de todos nuestros Cursos y Carreras para realizarlos de forma intensiva y personalizada ya sea en las Oficinas de COR TECH o in Company (Capital o Interior del País).
Enviando solamente un email a intensive@cortech.com.ar o llamando al (54)11-4312-7694.

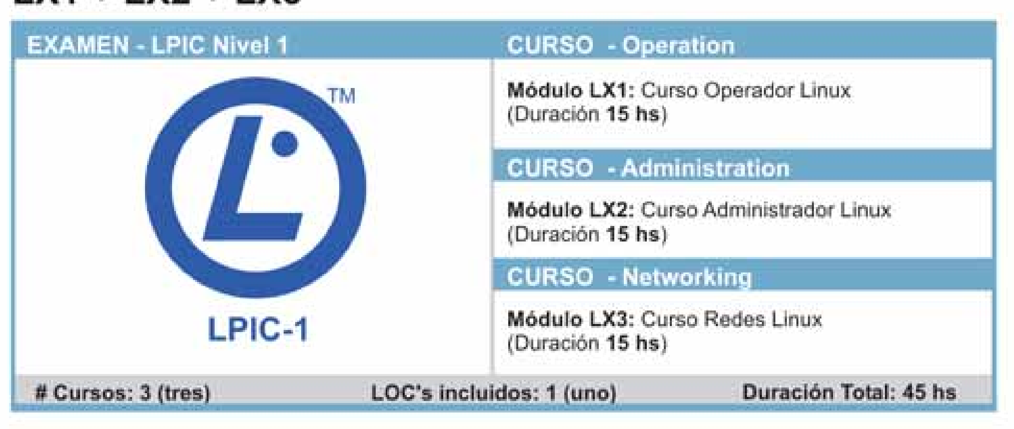
Certificaciones Macromedia

¿Dónde se pueden rendir los exámenes para certificarme como Macromedia Dreamweaver MX 2004 Designer, Developer y Flash MX 2004 Designer, Developer?

Podés hacer los exámenes en cualquier centro de tu localidad que provea exámenes VUE: Virtual Universities Enterprise (ver www.vue.com). Deberás entonces reservar tu turno de examen y abonar el costo correspondiente de 150.00 U$S para cada Examen MX 2004.

Más información respecto de las Certificaciones Macromedia MX 2004 podrás encontrarla en **www.macromedia.com**

Carrera Linux Completa
LX1 + LX2 + LX3



EXAMEN - LPIC Nivel 1	
LPIC-1	**CURSO - Operation** **Módulo LX1:** Curso Operador Linux (Duración **15 hs**)
	CURSO - Administration **Módulo LX2:** Curso Administrador Linux (Duración **15 hs**)
	CURSO - Networking **Módulo LX3:** Curso Redes Linux (Duración **15 hs**)

Cursos: 3 (tres) **LOC's incluidos: 1 (uno)** **Duración Total: 45 hs**









Carrera Linux Avanzada
LX1 + LX2 + LX3 + LX4 + LX5



EXAMEN - LPIC Nivel 2	
LPIC-2	**CURSO - Networking** **Módulo LX4:** Curso Redes Linux Avanzado (Duración **15 hs**)
	CURSO - Securing **Módulo LX5:** Curso Seguridad y Contra-Seguridad Linux (Duración **15 hs**)

Cursos: 5 (cinco) **LOC's incluidos: 2 (dos)** **Duración Total: 69 hs**

Carrera Linux Expert // Track Recomendado //
LX1 + LX2 + LX3 + LX4 + LX5 + 2 Workshops LX (a elección)



EXAMEN - LPIC Nivel 1 y Nivel 2	Workshops - Certificación
LPIC-1 LPIC-2	**LPIC-1:** Workshops para Exámenes LPI-101 y LPI-102 (Duración **12 hs**)
	LPIC-2: Workshops para Exámenes LPI-201 y LPI-202 (Duración **12 hs**)
EXAMEN - LPIC Nivel 3	**Workshops - Expert Linux**
LPIC-3	**LX6:** Workshops Servidor de Firewall y Squid (Comparación con ISA Server) (Duración **12 hs**)
	LX7: Workshops Clustering bajo Linux (Beowulf/ Open Mosix / Condor) (Duración **12 hs**)
	LX8: Workshops Implementando VPNs bajo Linux (FreeSwan) (Duración **12 hs**)
	LX9: Workshops Apache WEB Server (Duración **12 hs**)

Cursos: 7 (siete) **LOC's incluidos: 4 (cuatro)** **Duración Total: 93 hs**

Fechas Inicio Calendario
Carrera Linux Complete, Advanced y Expert

INICIO	DIAS	HORARIO
11-08-04	L-M-V	9.30 a 12.30
14-08-04	S	10.00 a 13.00
17-08-04	M-J	18.30 a 21.30
03-09-04	L-M-V	18.30 a 21.30
09-09-04	M-J	9.30 a 12.30
14-09-04	M-J	14.00 a 17.00
06-10-04	L-M-V	9.30 a 12.30
09-10-04	S	10.00 a 13.00
14-10-04	M-J	18.30 a 21.30
05-11-04	L-M-V	18.30 a 21.30
11-11-04	M-J	9.30 a 12.30
12-11-04	L-M-V	14.00 a 17.00
05-01-05	L-M-V	9.30 a 12.30
08-01-05	S	10.00 a 13.00
13-01-05	M-J	18.30 a 21.30
04-02-05	L-M-V	18.30 a 21.30
10-02-05	M-J	9.30 a 12.30
15-02-05	M-J	14.00 a 17.00
04-03-05	L-M-V	9.30 a 12.30
12-03-05	S	10.00 a 13.00
17-03-05	M-J	18.30 a 21.30
06-04-05	L-M-V	18.30 a 21.30
14-04-05	M-J	9.30 a 12.30
08-04-05	L-M-V	14.00 a 17.00
04-05-05	L-M-V	9.30 a 12.30
07-05-05	S	10.00 a 13.00
12-05-05	M-J	18.30 a 21.30
08-06-05	L-M-V	9.30 a 12.30
09-06-05	M-J	18.30 a 21.30
14-06-05	M-J	14.00 a 17.00
01-07-05	L-M-V	18.30 a 21.30
14-07-05	M-J	9.30 a 12.30
16-07-05	S	10.00 a 13.00

Para más información sobre la carrera Linux Completa, Avanzada y Expert visitá www.cortech.com.ar/lxc/lxc1.htm

Costos de las Carreras y Cursos

¿Dónde se puede averiguar el costo de los Cursos y Carreras Microsoft, Security, WEB Design y/o Linux?

Podés averiguar los costos de los Cursos y Carreras acercándote personalmente a COR Technologies SRL: Av. Córdoba 657 Piso 12, telefónicamente llamando al (54)11-4312-7694, vía correo electrónico a masinfo@cortech.com.ar, o en http://www.cortech.com.ar

http://www.cortech.com.ar/gen/Cursos y Fechas COR.pdf

Certificaciones LPI

¿Dónde se pueden rendir los exámenes para certificarme en LPIC 101, 102, 201 ó 202?

Podés hacer los exámenes en cualquier centro de tu localidad que provea exámenes VUE: Virtual Universities Enterprise (ver www.vue.com)

Deberás entonces reservar tu turno de examen y abonar el costo correspondiente de 150.00 U$S para cada Examen.

Más información respecto de las Certificaciones LPI podrás encontrarla en **www.lpi.org**